Diploma Thesis

Reduced Basis Method for the Shape Optimization of Racing Car Components

BY

Claudia Günther

Anna.Katharina.Claudia.Guenther@rwth-aachen.de
Matr.-Nr.: 252708

September 18, 2008

Supervised by
Dr. Gianluigi Rozza (EPFL)
Dr. Simone Deparis (EPFL)
Prof. Alfio Quarteroni (EPFL)
Prof. Marek Behr (RWTH)

Chair of Modeling and Scientific Computing (CMCS),
École Polytechnique Fédérale de Lausanne,
1015 Lausanne, Switzerland

Chair for Computational Analysis of Technical Systems (CATS),
Center for Computational Engineering Science (CCES),
RWTH Aachen University,
52056 Aachen, Germany

Affidavit:

I hereby declare that I wrote this thesis on my own and without the use of any other than the cited sources and tools and all explanations that I copied directly or in their sense are marked as such, as well as that the thesis has not yet been made available in neither in this nor in equal form to any other official commission.

Eidesstattliche Erklärung:

Ich versichere, dass ich diese Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie dass die Diplomarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Aachen, September 18, 2008 Claudia Günther

Acknowledgements

This thesis is the result of my semester abroad in Lausanne (Switzerland) at the Chair of Modelling and Scientific Computing (CMCS) of EPFL.
For their friendly encouragement I would like to thank all members of CMCS, especially my supervisors Dr. Gianluigi Rozza and Dr. Simone Deparis. I am very grateful for their availability, kindness and patience during my stay in Lausanne. They always offered me their help and our discussions and their helpful suggestions gave me many important and interesting ideas for this thesis. I must also thank Gianluigi Rozza for his dedicated and fast support and the many e-mails and priority mails during the revision of my thesis.
I thank Prof. Alfio Quarteroni for his willingness to supervise my thesis and his approval and kind support of my stay at EPFL.
Furthermore, I am very thankful to Prof. Marek Behr for the supervision of my thesis at the Chair for Computational Analysis of Technical Systems (CATS) of RWTH Aachen. Our correspondence and his motivating and valuable suggestions were an important contribution to the success of this work.

Contents

1 Introduction and Motivation

For the characterization, analysis and optimization of an engineering component or system, the rapid evaluation of certain system properties, such as maximum or average temperatures, heat transfer rates, flow rates, lift and drag forces or maximum and average stresses is a crucial ingredient. These outputs of interest can typically be obtained as algebraic functions of the solution of a parameter-dependent partial differential equation system. The parameters usually describe variations in loading or boundary conditions, material properties and geometries and thus specify a particular configuration of the system. The rapid and reliable evaluation of many of these input-output relationships require great computational expense and therefore strategies to reduce the computational effort and time have been developed. The reduced basis method is a very promising example for such a strategy. It is a reduced order method designed explicitly for the rapid, efficient and reliable solution of input-output relationships based on parametrized partial differential equations and is particularly suited and certified for the optimization of parametrized problems.

In this thesis, we will focus on the application of the reduced basis method for the shape optimization of racing car components. Shape optimization is a growing field in many different areas of engineering, which becomes more and more important as the computational resources are getting more and more fast and powerful. A traditional application of shape optimization is the field of structural analysis, where the aim is, for example, the optimization of the material costs of a component, while at the same time a certain stability level has to be ensured. Also in the domain of fluid dynamics, many different applications for shape optimization with the aim to optimize internal or external flow with respect to several different criteria can be found. Examples are the optimization of the design of blood pumps to minimize the blood damage, the optimization of the shape of reactors in process plants to maximize the mixing of different substances or the optimal design of wings and complete aircrafts to minimize the drag forces and maximize the achievable lift forces.

A similar goal is aimed at in the domain of automotive aerodynamics. The optimization of the shape of the whole car or of separate components shall ensure or increase the maneuverability and security of the vehicle while at the same time the aerodynamic drag forces have to be reduced as much as possible. For high performance vehicles such as Formula 1 race cars, the high velocities make it necessary to increase the contact pressure between the car and the ground, to ensure that the driver can steer, brake and accelerate also at high speed without loosing control over the car. To reach this aim, wings and flaps as attachment parts are used to introduce negative lift to the vehicle. The variations of these attachment parts concern their basic shape, their length and thickness, the rotation of the wing, the positioning on the car and the distance to the ground and to other parts of the car. Nevertheless there exist many different regulations for the specific competitions which have to be followed.

We will apply the reduced basis method to a set of simplified two-dimensional geometries which contain one or more parametrized airfoils. The race car itself and the interaction of the wings with the car or other attachment parts will not be modeled. We will use several different fluid models of increasing complexity to model the flow around the airfoils. Our analysis will focus on the behavior and performance of the reduced basis method for the different models and geometries. The special abilities of the reduced basis method in the domain of shape optimization will be highlighted and demonstrated by means of several examples.

In chapter 2, we will first introduce the theory of the reduced basis method necessary for the

following investigations. A short review of the development of the reduced basis method and its historical background will also be provided. Then in chapter 3, we present the software package rbMIT designed for the solution of parametrized partial differential equations with the reduced basis method used for all computational examples in this work. Additionally, the set of parametrized geometries and fluid flow models to which the reduced basis method shall be applied are introduced in the same chapter. The content of chapter 4 is the application of the reduced basis method to potential flows around an airfoil, while chapter 5 concentrates on thermal flows around the airfoil, governed by a convection-diffusion equation. Chapter 6 and chapter 7 both consider the application of the reduced basis method to viscous flows governed by the Stokes equations on the set of parametrized geometries. Chapter 6 thereby uses a penalty method to simplify the governing equations, while chapter 7 provides a short perspective on the solution of the standard Stokes system by the reduced basis method. Finally in chapter 8, the thesis concludes with a summary of the results that we obtained and some suggestions for future work.

2 Fundamentals of the Reduced Basis Method for Parametrized Coercive Elliptic PDEs

In this chapter, we will introduce the theory of the reduced basis (RB) method. First, we will introduce the problem classes for which the reduced basis method can have significant advantages. Then a short summary of the historical background and recent developments of the reduced basis method will be given. In the following, the relevant steps for the generation of the rapidly converging global RB approximation spaces for the approximation of the solution of parametrized coercive elliptic partial differential equations with affine parameter dependence will be explained. The affine parameter dependence enables an efficient offline-online computational decomposition. *A posteriori* error estimators and lower bounds for the coercivity constant will play an important role in this process. Finally, the possible (affine) geometric variations used in this work and their application in the reduced basis context will be detailed.

2.1 Introduction

An increasing number of engineering analyses require the solution of partial differential equations (PDEs). Most of them include the solution of not only one but many different configurations of the system. Usually, the computational costs for the solution even of relatively simple cases are very high and the solution of many problems remains unaffordable - although the computational performances have considerably increased in the past few years. It is necessary to develop techniques that are able to reduce the complexity of the system without a loss of information or accuracy of the results. The reduced basis method is a promising approach to fill this gap as it allows not only the *rapid* and *efficient*, but also the *reliable* solution of partial differential equations.

2.1.1 The Input-Output Relation

In many of these applications, the main goal is not the solution of the underlying PDE for the *field variables*, but the evaluation of *input-output relationships*. Here, the *output* is expressed as a functional of the field variable and can be, for example, the mean temperature in the domain, a mass flow rate over a domain boundary or - in our case - the aerodynamic lift and/or drag forces. The *input-parameter* vector identifies a particular configuration of the system. Usually, this includes geometric variations, physical properties and characteristics as well as boundary/initial conditions and sources. The *field variable* as solution of the PDE connects the input parameters and the outputs of interest.

2.1.2 The Many-Query and Real-Time Contexts

The reduced basis method allows us to reduce the online computational time (both of the field solution and of the outputs of interest) notably. This advantage is gained by additional offline effort. Therefore, the methodology presented in this work is suited particularly for two different classes of problems, where the tradeoff between (sometimes considerably) increased offline computational times and strongly improved online performance can be accepted. These problems either arise in

the *real-time context* or in the *many-query context*. For both these problem classes, the online performance is extremely important while increased offline effort is less critical. The real-time context arises in a wide range of applications, e.g. for control engineering and in parameter estimation problems. Examples for the many-query context are the multi-model/multi-scale simulation or design optimization where the online performance is critical because the solution of a huge number of problems - up to the tens of thousands input-output relations - is necessary, which makes the additional offline effort affordable.

It is important to note that the RB method should not and can not replace "classical" numerical techniques such as the finite element (FE) method. This is due to two reasons. First, the application of the RB method to problems that require the solution only for one or a few configurations would be inefficient. The reduced online complexity would not balance the large offline effort needed for the application of the RB method compared to the FE method (or other solution methods). In addition, the RB method is not an independent method. It has to be built upon another method for the solution of the particular problem. In fact, the RB method tries to approximate the solution of the system that would be obtained by applying the given underlying solution method to the problem, and not the exact solution. In this work, we will always consider a finite element discretization as underlying solution method but also other methods (e.g. finite volume or spectral methods) would be theoretically possible.

2.1.3 Historical Background

A brief introduction to the historical background and the more recent developments of the reduced basis method shall be given here. For a more detailed presentation see [55, 64] and [68].

The reduced basis method for single parameter problems was first introduced in the late 1970s by Almroth, Stern and Brogan in the domain of nonlinear structural analysis [2]. The method has been developed further by Noor in the following years [47, 43, 44, 48, 49, 50, 46, 45] and extended to multi-parameter problems. A first *a priori* error analysis for single parameter problems has been carried out by Fink and Rheinboldt [16, 17]. Further work that concentrated on *a priori* error analysis and on different approximation spaces has been done by Porsching [57]. In the 1990s, this topic has been investigated again by Rheinbolt [63] and by Barrett and Reddien [8]. The reduced basis method has been applied to different classes of problems: viscous fluid flow and the Navier-Stokes equations [56], fluid control problems [25, 26, 27, 28], ordinary differential equations [58] and differential algebraic equations [31]. These early methods were typically rather local and low-dimensional in parameter. In [6], Balmes was the first to apply the RB method to general multi-parameter problems.

More recently, the need for reliable *a posteriori* error estimators has led to a number of works on this topic [59, 35, 36, 34, 77, 64, 72]. Much effort is also devoted to effective sampling strategies for global approximation spaces also for higher dimensional parameter domains [11, 41, 66]. In the past few years, this methodology has been applied to a wide range of problems including elliptic as well as parabolic and hyperbolic problems. Problems came from the domains of elasticity (with fracture), acoustics, quantum mechanics and quantum chemistry, but also from homogenization, Boltzmann models, environmental engineering and inverse problems (for actual works on these topics refer to [5]). Some examples for applications in fluid flow problems (e.g. [67, 69, 76] and in optimization and optimal control (e.g. [19, 52, 61, 67]) shall be mentioned here as this thesis focuses on these topics. A significant part of this recent work in the field of the reduced basis methodology and its applications has been done by A.T. Patera and his group at the Department of Mechanical Engineering at Massachusetts Institute of Technology. An overview of their work and recent advancements is given on [5]. Two recent publications give a very comprehensive summary of the methodology of the reduced basis method developed so far for coercive elliptic PDEs with affine parameter dependence [55, 68].

2.2 Problem Definition

In the following section, we will state the abstract problem for coercive elliptic partial differential equations with affine parameter dependence. The methodology of the RB method described further in this work will apply to this class of problems.

2.2.1 Exact Statement

We consider a suitably regular (smooth) domain $\Omega \subset \mathbb{R}^d$, $d = 1$, 2, or 3 with Lipschitz-continuous boundary $\partial\Omega$, where d is the spacial dimension. X^e is an associated (infinite dimensional) Hilbert space satisfying $(H_0^1(\Omega))^\nu \subset X^e \subset (H^1(\Omega))^\nu$ with $\nu = 1$ for a scalar problem and $\nu = d$ for a vectorial problem. Here, $H^1(\Omega) = \{v \in L^2(\Omega) | \nabla v \in (L^2(\Omega))^d\}$, $H_0^1(\Omega) = \{v \in H^1(\Omega) |\, v|_{\partial\Omega} = 0\}$, and $L^2(\Omega) = \{v \text{ measurable} | \int_\Omega v^2 \text{ finite}\}$. The inner product and norm associated with X^e are given by $(\cdot,\cdot)_{X^e}$ and $\|\cdot\|_{X^e} = (\cdot,\cdot)_{X^e}^{1/2}$, respectively. Additionally, we define an input parameter domain $\mathcal{D} \subset \mathbb{R}^P$. The superscript ^{e} refers to "exact".

The "exact" problem is then given as: For any given parameter $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$, evaluate the scalar output of interest

$$s^e(\boldsymbol{\mu}) = l(u^e(\boldsymbol{\mu})),$$

where the field variable $u^e(\boldsymbol{\mu})$ satisfies

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \qquad \forall v \in X^e. \tag{2.1}$$

We assume that the form $a(\cdot,\cdot;\boldsymbol{\mu}) : X^e \times X^e \to \mathbb{R}$ is bilinear, coercive and continuous, and $l(\cdot) : X^e \to \mathbb{R}$ is a bounded linear functional. If the bilinear form is symmetric and the equation $l = f$ holds, we say that our problem is "compliant".

2.2.2 "Truth" Finite Element Approximation

We proceed now to the finite element approximation of (2.1) (see e.g. [15, 62]). We replace X^e with $X^{N_t} \subset X^e$ which is a sequence of (conforming) "truth" finite element approximation spaces of finite but large dimension N_t. The inner product and norm associated with X^{N_t} are inherited from X^e. We will define them below.

The "truth" finite element approximation is then given as: For any given parameter $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$, evaluate

$$s^{N_t}(\boldsymbol{\mu}) = l(u^{N_t}(\boldsymbol{\mu})),$$

where $u^{N_t}(\boldsymbol{\mu})$ satisfies

$$a(u^{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \qquad \forall v \in X^{N_t}. \tag{2.2}$$

Our "truth" FE approximation $u^{N_t}(\boldsymbol{\mu}) \in X^{N_t}$ to $u^e(\boldsymbol{\mu})$ is thus defined as the Galerkin projection of $u^e(\boldsymbol{\mu})$ onto X^{N_t}.

The finite element discretization shall be assumed to be sufficiently rich such that $u^{N_t}(\boldsymbol{\mu})$ and $s^{N_t}(\boldsymbol{\mu})$ are sufficiently close to $u^e(\boldsymbol{\mu})$ resp. $s^e(\boldsymbol{\mu})$ - this is the reason why we call it "truth" approximation. The RB field solution and RB output shall approximate this "truth" finite element field solution $u^{N_t}(\boldsymbol{\mu})$ and output $s^{N_t}(\boldsymbol{\mu})$ and not the "exact" solutions $u^e(\boldsymbol{\mu})$ and $s^e(\boldsymbol{\mu})$. The reduced basis error will thus be evaluated with respect to the "truth" finite element solutions. We must be certain that our method remains computationally stable and efficient as $N_t \to \infty$.

We will define two different inner products and norms for members of X^{N_t} and X^e. First, our

energy inner product and energy norm are defined as

$$(w, v)_{\boldsymbol{\mu}} \equiv a(w, v; \boldsymbol{\mu}), \qquad \forall w, v \in X \tag{2.3}$$

$$\|w\|_{\boldsymbol{\mu}} \equiv (w, w)_{\boldsymbol{\mu}}^{1/2}, \qquad \forall w \in X. \tag{2.4}$$

A second inner product and norm, the X^{N_t} resp. X^e inner product and norm, shall be defined as follows: for given $\overline{\boldsymbol{\mu}} \in \mathcal{D}$ and (non-negative) real τ,

$$(w, v)_X \equiv (w, v)_{\overline{\boldsymbol{\mu}}} + \tau (w, v)_{L^2(\Omega)}, \quad \forall w, v \in X \tag{2.5}$$

$$\|w\|_X \equiv (w, w)_X^{1/2}, \qquad \forall w \in X. \tag{2.6}$$

The L^2-norm is defined as $(w, v)_{L^2(\Omega)} \equiv \int_\Omega wv$.

2.2.3 Well Posedness

We define our exact and FE coercivity constants as

$$\alpha^e(\boldsymbol{\mu}) = \inf_{w \in X^e} \frac{a(w, w; \boldsymbol{\mu})}{\|w\|_X^2}, \tag{2.7}$$

and

$$\alpha^{N_t}(\boldsymbol{\mu}) = \inf_{w \in X^{N_t}} \frac{a(w, w; \boldsymbol{\mu})}{\|w\|_X^2}. \tag{2.8}$$

As we assumed that our bilinear form is coercive and our FE approximation spaces are conforming, it follows that $\alpha^{N_t}(\boldsymbol{\mu}) \geq \alpha^e(\boldsymbol{\mu}) \geq \alpha_0 > 0$, $\forall \boldsymbol{\mu} \in \mathcal{D}$. The continuity constants are defined similarly as

$$\gamma^e(\boldsymbol{\mu}) = \sup_{w \in X^e} \sup_{v \in X^e} \frac{a(w, v; \boldsymbol{\mu})}{\|w\|_X \|v\|_X} \tag{2.9}$$

and

$$\gamma^{N_t}(\boldsymbol{\mu}) = \sup_{w \in X^{N_t}} \sup_{v \in X^{N_t}} \frac{a(w, v; \boldsymbol{\mu})}{\|w\|_X \|v\|_X}. \tag{2.10}$$

It is clear from our continuity and conforming hypotheses that $\gamma^{N_t}(\boldsymbol{\mu}) \leq \gamma^e(\boldsymbol{\mu}) \leq \gamma_0 < \infty$, $\forall \boldsymbol{\mu} \in \mathcal{D}$.

If the bilinear form a fulfills these requirements of coercivity and continuity and if the linear form f is bounded, the system (2.1) has a unique solution.

2.2.4 Affine Parameter Dependence

We shall also make an important assumption on the nature of the parametric dependence of our problem. In particular, we shall suppose that the parametric bilinear form a is "affine" in the parameter $\boldsymbol{\mu}$. This means that it it can be expressed as

$$a(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(w, v), \quad \forall w, v \in X^{N_t}, \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{2.11}$$

Here, Q_a should be a finite and preferably small integer. The functions $\Theta_a^q(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$ depend on $\boldsymbol{\mu}$ and are typically very smooth, while the bilinear forms $a^q(\cdot, \cdot) : X^e \times X^e \to \mathbb{R}$ are $\boldsymbol{\mu}$-independent X^e-continuous bilinear forms. The linear form f may also depend affinely on the parameter and can in this case be expressed as a sum of Q^f products of parameter-dependent functions and parameter-independent X^e-bounded linear forms.

The assumption of affine parameter dependence is crucial for the computational performance

of our method as it allows an efficient offline-online decomposition of the relevant computational procedures. In fact, this assumption is not too restrictive as there exist many applications with both geometric and property variations which exhibit an affine dependence on the parameter. Some examples are shown in the practical part of this work. Here, we shall only consider the affine case, but the methodology can also be extended (with some performance losses and more complicated procedures, of course) to the non-affine case. For references, see [7, 19, 40, 66].

2.3 Reduced Basis Approximation

The parametric real-time and many-query settings introduced before, that are suited for the application of reduced basis methods, represent two different computational opportunities. The reduced basis methodology that we will present in the following is constructed in a way that gainfully exploits these opportunities. The first opportunity is the fact that in the parametric setting our attention can be restricted to a typically smooth and rather low-dimensional parametrically induced manifold, which is much smaller than the unnecessarily rich generic approximation spaces of e.g. the FE method. The second opportunity is - as mentioned before - that an increased offline time can be accepted in these contexts if the online evaluation time of the input-output relation is reduced considerably in exchange. In the following, we will describe the main steps of the reduced basis method presented in [55, 68]. Special attention will be paid to the question how the methodology consequently makes use of these two opportunities

2.3.1 Manifold of Solutions

We will first explain in more detail the first opportunity and its implications on an efficient methodology. As noted above, the field variable $u^e(\boldsymbol{\mu})$ is not an arbitrary member of the infinite-dimensional solution space X^e associated with the underlying partial differential equation. In fact, it resides on a much lower-dimensional and typically smooth manifold $M \equiv \{u^e(\boldsymbol{\mu})|\boldsymbol{\mu} \in D\}$ induced by the parametric dependence. For example, in the case of a single parameter $\boldsymbol{\mu} \in D \subset \mathbb{R}^{P=1}$, $u^e(\boldsymbol{\mu})$ describes a one-dimensional filament that winds through X^e. This situation is depicted in Figure 2.1. Thus, the possible solutions $u^e(\boldsymbol{\mu})$ do not cover the entire space X^e, which means that this space is too general as it can represent a much wider range of functions.

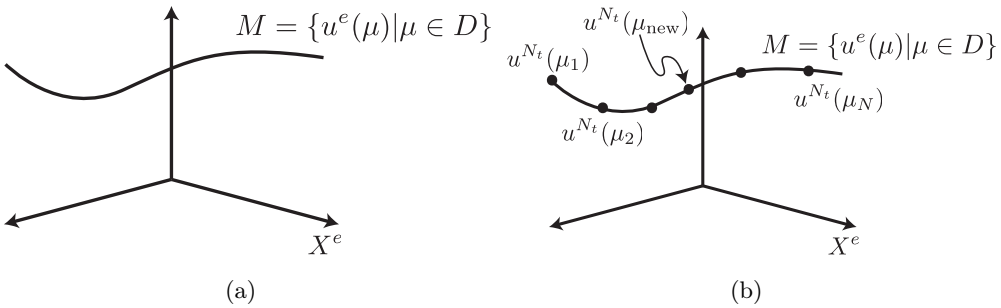


Figure 2.1: (a) Low-dimensional manifold on which the field variable resides and (b) approximation of a new solution at $\boldsymbol{\mu}_{\text{new}}$ with the "snapshots" $u^{N_t}(\boldsymbol{\mu}_n)$, $1 \leq n \leq N$.

The "truth" finite element approximation space X^{N_t} is constructed to approximate all members of X^e. It is therefore still much too general as it includes many functions that do not reside on the manifold of interest M. To approximate $u^{N_t}(\boldsymbol{\mu})$ by an adjusted method, it is sufficient to be able to approximate all functions that lie on M while it is not necessary to represent every single function in X^{N_t}. If we exploit this observation and restrict our attention and adjusted approximation space

to the parameter-induced low-dimensional solution manifold, we can effect substantial dimension reduction and considerable computational economies.

The basic idea is to construct a special approximation space for the manifold M by using the pre-computed solutions $u^{N_t}(\boldsymbol{\mu_n})$ at $N \ll N_t$ selected points $\boldsymbol{\mu}_n$ along M, as shown in Figure 2.1. The solution $u^N(\boldsymbol{\mu}_{\text{new}})$ at a newly chosen point $\boldsymbol{\mu}_{\text{new}}$ can then be approximated by taking an appropriate linear combination of the sample points $u^{N_t}(\boldsymbol{\mu}_n)$, $n = 1..N$, that means by a projection onto the adjusted approximation space. We will now precise this idea.

2.3.2 RB Spaces

Our starting point is the FE approximation space of dimension N_t, X^{N_t}. We then want to construct an associated sequence of hierarchical approximation spaces with maximum dimension $N_{\max}$, $X_N^{N_t}$, $N = 1, .., N_{\max}$. Each of these spaces is an N-dimensional subspace of X^{N_t}. Hierarchical means that the spaces fulfill the following relationship:

$$X_1^{N_t} \subset X_2^{N_t} \subset \ldots \subset X_{N_{\max}}^{N_t} \subset X^{N_t}. \tag{2.12}$$

This quality will play an important role for computational and memory efficiency.

In this work, we will only use Lagrange reduced basis spaces [57]. It is possible to work with Taylor [47, 57] and Hermite [26] spaces as well and much of the methodology does not change for these spaces. We first introduce - for given $N \in \{1, \ldots, N_{\max}\}$ - a set of nested samples in parameter space,

$$S_N = \{\boldsymbol{\mu}_1 \in D, \ldots, \boldsymbol{\mu}_N \in D\}, \quad 1 \leq N \leq N_{\max}, \tag{2.13}$$

such that $S_1 \subset \cdots \subset S_N \subset S_{N_{\max}}$ (the parameter samples are nested). The associated Lagrange RB approximation spaces are then given as

$$W_N^{N_t} = \text{span}\{u^{N_t}(\boldsymbol{\mu}_n), \quad 1 \leq n \leq N\}, \quad 1 \leq N \leq N_{\max}, \tag{2.14}$$

where $u^{N_t}(\boldsymbol{\mu}_n) \in X^{N_t}$ is the solution to (2.2) for $\boldsymbol{\mu} = \boldsymbol{\mu}_n$. By construction, the Lagrange RB spaces are hierarchical: $W_1^{N_t} \subset W_2^{N_t} \subset \cdots \subset W_{N_{\max}}^{N_t} \subset X^{N_t}$.

The $u_n^{N_t} = u^{N_t}(\boldsymbol{\mu}_n)$, $1 \leq n \leq N_{\max}$ are often called "snapshots" or, more precisely, "retained snapshots" of the parametric manifold M. The next question we have to address is the question how we can choose a good combination of the retained snapshots to approximate the solution for a new parameter value and how we can build a stable RB basis out of the retained snapshots.

2.3.3 Galerkin Projection

Given our hypotheses on a and f, standard Galerkin projection is the best choice to obtain the reduced basis approximation $u_N^{N_t}(\boldsymbol{\mu})$: For any $\boldsymbol{\mu} \in D$, $u_N^{N_t}(\boldsymbol{\mu}) \in W_N^{N_t}$ satisfies

$$a(u_N^{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \qquad \forall v \in W_N^{N_t}. \tag{2.15}$$

We then evaluate

$$s_N^{N_t}(\boldsymbol{\mu}) = l(u_N^{N_t}(\boldsymbol{\mu})). \tag{2.16}$$

In theory, we can choose for the Lagrange sample points (2.13) any set of parameter values that induce a linearly independent set of retained snapshots (2.14). However, the snapshots will become more and more collinear as N increases because of the rapid convergence of the Lagrange space: If $W_N^{N_t}$ is already able to approximate well any member of M, then the next snapshot $u_{N+1}^{N_t}(\boldsymbol{\mu})$ will be "almost" linearly dependent of the members of the "old" space $W_N^{N_t}$. The direct choice of the retained snapshots as basis functions would therefore lead to a very poorly conditioned equation system for the solution of (2.15).

To create a well-conditioned set of basis functions, we will therefore apply the Gram-Schmidt process in the $(\cdot,\cdot)_X$ inner product to our set of snapshots $u_N^{N_t}(\boldsymbol{\mu}_n)$, $1 \le n \le N_{\max}$. This process will return us a mutually orthogonal set of basis functions $\zeta_n^{N_t}$, $1 \le n \le N_{\max}$ which we can choose as our bases for $W_N^{N_t}$ resp. $X_N^{N_t}$. These basis functions will fulfill the orthogonality condition

$$(\zeta_n^{N_t}, \zeta_m^{N_t})_X = \delta_{n,m}, \qquad 1 \le n, m \le N_{\max}, \tag{2.17}$$

where $\delta_{n,m}$ is the Kronecker-delta symbol. This orthogonality condition is necessary to ensure a well-conditioned reduced basis algebraic system.

The orthogonalization process is given as follows [55]:

$$\begin{aligned}
&\zeta_1^{N_t} = u_1^{N_t} / \|u_1^{N_t}\|_X; \\
&\texttt{for } n = 2 : N_{\max} \\
&\quad z_n^{N_t} = u_n^{N_t} - \sum_{m=1}^{n-1} (u_n^{N_t}, \zeta_m^{N_t})_X \zeta_m^{N_t}; \\
&\quad \zeta_n^{N_t} = z_n^{N_t} / \|z_n^{N_t}\|_X; \\
&\texttt{end.}
\end{aligned} \tag{2.18}$$

The discrete equations associated to the Galerkin system (2.15) are then constructed by inserting the expansion of $u_N^{N_t}(\boldsymbol{\mu})$ in the basis functions

$$u_N^{N_t}(\boldsymbol{\mu}) = \sum_{m=1}^{N} u_{N\,m}^{N_t}(\boldsymbol{\mu}) \zeta_m^{N_t} \tag{2.19}$$

and $v = \zeta_n^{N_t}$, $1 \le n \le N$ into (2.15). The equation obtained is the reduced basis stiffness equation for the reduced basis coefficients $u_{N\,m}^{N_t}(\boldsymbol{\mu})$, $1 \le m \le N$:

$$\sum_{m=1}^{N} a(\zeta_m^{N_t}, \zeta_n^{N_t}; \boldsymbol{\mu}) u_{N\,m}^{N_t}(\boldsymbol{\mu}) = f(\zeta_n^{N_t}), \quad 1 \le n \le N. \tag{2.20}$$

The reduced basis output prediction can then be evaluated as

$$s_N^{N_t}(\boldsymbol{\mu}) = \sum_{m=1}^{N} u_{N\,m}^{N_t}(\boldsymbol{\mu}) l(\zeta_m^{N_t}). \tag{2.21}$$

As shown in [55], the condition number of the matrix $a(\zeta_m^{N_t}, \zeta_n^{N_t}; \boldsymbol{\mu})$, $1 \le n, m \le N$ is independent of N and N_t and bounded by $\gamma^e(\boldsymbol{\mu})/\alpha^e(\boldsymbol{\mu})$.

2.3.4 Offline-Online Procedure

The System (2.20) has normally a very small size compared to the system that arises from standard FE discretization of (2.2): The system (2.20) consists of a set of N linear algebraic equations in N unknowns, while the FE discretization would lead to a set of $N_t \gg N$ equations in $N_t \gg N$ unknowns. Nevertheless, the elements of $W_N^{N_t}$, the basis functions $\zeta_n^{N_t}$, $1 \le n \le N$, are associated with the underlying FE space and thus N_t-dependent. This makes the formation of the stiffness matrix and the load vector for our RB system (2.20) for every new value of $\boldsymbol{\mu}$ N_t-dependent, even though the solution of this system is not. To eliminate this N_t-dependency, which would lead to a very poor online performance, we construct a very efficient offline-online procedure. This procedure is based on the affine parameter dependence, as we now discuss. Equation (2.11) allows

us to express our system (2.20) as

$$\sum_{m=1}^{N}\left(\sum_{q=1}^{Q_a}\Theta^q(\boldsymbol{\mu})a^q(\zeta_m^{N_t},\zeta_n^{N_t})\right)u_{N\,m}^{N_t}(\boldsymbol{\mu})=f(\zeta_n^{N_t}),\quad 1\le n\le N. \tag{2.22}$$

If f is also affinely dependent on the parameter $\boldsymbol{\mu}$, the right hand side of this equation can be expanded in a similar sum of Q_f affine terms, but in the following discussion we will assume that f does not depend on $\boldsymbol{\mu}$.

We see that the terms dependent on the $\zeta_n^{N_t}$ are now separated of those dependent on the parameter. Therefore we can precompute all terms independent of the parameter in the offline stage and thus the operation count for the online computations is independent of N_t.

In the offline stage - performed only once - we first compute the $u^{N_t}(\boldsymbol{\mu})$, $1\le n\le N_{\max}$ and form the $\zeta_n^{N_t}$, $1\le n\le N_{\max}$. After that, the

$$f(\zeta_n^{N_t}),\quad 1\le n\le N_{\max} \tag{2.23}$$

and

$$a^q(\zeta_m^{N_t},\zeta_n^{N_t}),\quad 1\le n,m\le N_{\max},\quad 1\le q\le Q_a \tag{2.24}$$

can be computed and stored. The offline operation count thus depends on N^{N_t}, Q_a and $N_{\max}$. The storage needed can be reduced to $\mathcal{O}(Q_aN_{\max}^2)$.

In the online stage - performed many times, for each new value of $\boldsymbol{\mu}$ - we use the precomputed matrix (2.24) to assemble the (full) $N\times N$ stiffness matrix

$$\sum_{q=1}^{Q_a}\Theta^q(\boldsymbol{\mu})a^q(\zeta_m^{N_t},\zeta_n^{N_t}),\quad 1\le n,m\le N. \tag{2.25}$$

We then solve the resulting system (2.22) to obtain the $u_{N,\,m}^{N_t}(\boldsymbol{\mu})$, $1\le m\le N$ and evaluate the output approximation (2.21). The operation count for the online stage is then $\mathcal{O}(Q_aN^2)$ to assemble (2.24), $\mathcal{O}(N^3)$ to invert the stiffness matrix (remember that it is full) and $\mathcal{O}(N)$ to evaluate the inner product for the output computation. Thanks to the hierarchical condition , the online storage is only $\mathcal{O}(Q_aN_{\max}^2)+\mathcal{O}(N_{\max})$, as for any given N the necessary RB $N\times N$ matrices may be extracted of the corresponding "maximum" $N_{\max}\times N_{\max}$ system.

The essential point is that our online computational costs are dependent on Q_a and N, but independent of N_t. Since $N\ll N_t$, we can expect significant (orders of magnitude) speedup in the online stage compared to the pure FE approach. This implies also that we may choose N_t very large in order to eliminate the error between the exact solution and the FE predictions without effecting the reduced basis online efficiency. In fact, the bigger the underlying FE system and thus N_t is chosen, the bigger the speedup by the use of the RB method in the online stage will be. However, we should keep in mind that the tribute has to be paid in the offline phase which is still N_t-dependent.

2.3.5 Sampling Strategy: A "Greedy" Algorithm

The question we deal with in this section is how to choose the sample points $\boldsymbol{\mu}_n$, $1\le n\le N$ for a given N in an optimal way, that means such that the accuracy of the resulting RB approximation is maximized. The key ingredient is here a rigorous, sharp and inexpensive a posteriori error bound $\Delta_N^{N_t}(\boldsymbol{\mu})$ such that

$$\|u^{N_t}(\boldsymbol{\mu})-u_N^{N_t}(\boldsymbol{\mu})\|_X\le\Delta_N^{N_t}(\boldsymbol{\mu}) \tag{2.26}$$

for all $\boldsymbol{\mu} \in D$ and for all N. The properties sharpness and rigor can be quantified by introducing the effectivity

$$\eta_N^{N_t}(\boldsymbol{\mu}) \equiv \frac{\Delta_N^{N_t}(\boldsymbol{\mu})}{\|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_X}. \tag{2.27}$$

Rigor means that we require that $\Delta_N^{N_t}(\boldsymbol{\mu})$ is never less than the true error. Sharpness means that we require that $\Delta_N^{N_t}(\boldsymbol{\mu})$ is not too much larger than the true error. To put this into one equation, $\Delta_N^{N_t}(\boldsymbol{\mu})$ must fulfill $1 \leq \eta_N^{N_t}(\boldsymbol{\mu}) \leq \eta_{\max,\,\mathrm{UB}}, \forall \boldsymbol{\mu} \in D, 1 \leq N \leq N_{\max}$, where $\eta_{\max,\,\mathrm{UB}}$ is finite (preferably close to 1) and independent of N. The computation of the error bound is "inexpensive" if we can compute $\boldsymbol{\mu} \to \Delta_N^{N_t}(\boldsymbol{\mu})$ extremely fast, which implies that in the limit of many evaluations the marginal cost is independent of N_t. We discuss the construction and properties of such an error estimate in detail in section 2.4. We will now proceed to the "greedy" procedure that makes use of this a posteriori error estimate to construct hierarchical Lagrange RB approximation spaces.

We are given $N_{\max}$, which can be set either directly or through a prescribed error tolerance ϵ_{tol}, and a training sample $\Xi_{\mathrm{train}} \subset D$ (a very fine sample of $n_{\mathrm{train}} = |\Xi_{\mathrm{train}}|$ points in the parameter domain). This "training" sample shall serve as surrogate for D in the following generation of the reduced basis space and the choice of n_{train} and Ξ_{train} therefore has important offline and online computational implications. We then choose at random $\boldsymbol{\mu}_1 \in \Xi_{\mathrm{train}}$, the first sample point to be added to the Lagrange parameter samples $S_1 = \{\boldsymbol{\mu}_1\}$, and set $W_1^{N_t\,\mathrm{Greedy}} = \mathrm{span}\{u^{N_t}(\boldsymbol{\mu}_1)\}$.

The algorithm proceeds as follows:

$$\begin{aligned}
&\texttt{for } N = 2 : N_{\max} \\
&\quad \boldsymbol{\mu}_N = \arg \max_{\boldsymbol{\mu} \in \Xi_{\mathrm{train}}} \Delta_{N-1}^{N_t}(\boldsymbol{\mu}); \\
&\quad \epsilon_{N-1} = \Delta_{N-1}^{N_t}(\boldsymbol{\mu}_N); \\
&\quad \texttt{if } \epsilon_{N-1} \leq \epsilon_{\mathrm{tol}} \\
&\qquad N_{\max} = N - 1; \\
&\quad \texttt{end}; \\
&\quad S_N^{N_t} = S_{N-1} \cup \boldsymbol{\mu}_N, \\
&\quad W_N^{N_t\,\mathrm{Greedy}} = W_{N-1}^{N_t\,\mathrm{Greedy}} + \mathrm{span}\{u^{N_t}(\boldsymbol{\mu}_N)\} \\
&\texttt{end}.
\end{aligned} \tag{2.28}$$

Hence, the greedy algorithm chooses in each iteration N that particular candidate snapshot (over all candidate snapshots $u^{N_t}(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \Xi_{\mathrm{train}}$) that is least well approximated by the projection on the "old" RB space $W_{N-1}^{N_t\,\mathrm{Greedy}}$ and appends it to the retained snapshots. The most crucial point of this strategy is that the error is not measured by the (very expensive) "true" error $\|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_X$ but by the inexpensive a posteriori error bound $\Delta_N^{N_t}(\boldsymbol{\mu})$. In doing so we need to compute only the $N_{\max}$ (typically very few) FE retained snapshots[1]. This permits us to perform offline a very exhaustive search for the best sample with n_{train} very large and thus get most rapidly uniformly convergent spaces $W_N^{N_t\,\mathrm{Greedy}}$. Online, we can exploit the low marginal cost of the error estimate and the hierarchical condition of the $W_N^{N_t\,\mathrm{Greedy}}$, $1 \leq N \leq N_{\max}$ to determine the smallest N (the most efficient approximation) for which we rigorously achieve the desired accuracy.

2.3.6 Convergence Analysis

We will now state some theoretical evidence that the reduced basis approximation indeed converges to the FE approximation, if a good low-dimensional approximation space exists and consequently a

[1] This is contrary to the POD approach, where we have to compute the FE solutions for all members of Ξ_{train} as we measure the error by the true error $\|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_X$, see [55, 68].

good Lagrange RB approximation space can be constructed by the greedy algorithm. In addition, results shall be presented that confirm the existence of suited RB approximation spaces and provide upper bounds for convergence rates in terms of the given data for the problem.

Optimality

First, the classical Galerkin optimality result for the projection in the RB approximation space shall be demonstrated:

$$\|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} = \inf_{w_N \in X_N^{N_t}} \|u^{N_t}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}, \tag{2.29}$$

$$\|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_X = \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha^e(\boldsymbol{\mu})}} \inf_{w_N \in X_N^{N_t}} \|u^{N_t}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\|_X, \tag{2.30}$$

and for the compliant case also

$$\begin{aligned} s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu}) &= \|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2 \\ &= \inf_{w_N \in X_N^{N_t}} \|u^{N_t}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2, \end{aligned} \tag{2.31}$$

as well as

$$0 < s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu}) \leq \gamma^e(\boldsymbol{\mu}) \inf_{w_N \in X_N^{N_t}} \|u^{N_t}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\|_X^2. \tag{2.32}$$

To prove (2.29), we first state, since our reduced basis space is conforming, $X_N^{N_t} \subset X^{N_t}$, the Galerkin orthogonality:

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = a(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = 0, \quad \forall v \in X_N^{N_t}. \tag{2.33}$$

It then follows that for any $w_N = u_N^{N_t} + v_N \in X_N^{N_t}$ $(v_N \neq 0)$,

$$\begin{aligned} &a(u^{N_t}(\boldsymbol{\mu}) - w_N, u^{N_t}(\boldsymbol{\mu}) - w_N; \boldsymbol{\mu}) \\ &\quad = a(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}) - v_N, u^{N_t} - u_N^{N_t}(\boldsymbol{\mu}) - v_N; \boldsymbol{\mu}) \\ &\quad = a(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}), u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}); \boldsymbol{\mu}) - 2a(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) + a(v_N, v_N; \boldsymbol{\mu}) \\ &\quad \geq a(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}), u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}); \boldsymbol{\mu}) \end{aligned} \tag{2.34}$$

from (2.33), symmetry of a and coercivity (2.8). Inequality (2.29) in the energy norm then follows directly. To obtain the result in the X-norm (2.30), we apply the energy-norm bound (2.29) together with coercivity and continuity. The output result (2.31) is found by invoking compliance and Galerkin orthogonality ($s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu}) = f(e(\boldsymbol{\mu})) = a(u^{N_t}(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) = a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu})$) and then using again the energy-norm bound (2.29). The result (2.32) follows from (2.31) and continuity.

The output approximation $s_N^{N_t}(\boldsymbol{\mu})$ thus converges to $s^{N_t}(\boldsymbol{\mu})$ as the square of the error in the field variable $u_N^{N_t}$. We also note that $s_N^{N_t}(\boldsymbol{\mu})$ is a lower bound for $s^{N_t}(\boldsymbol{\mu})$.

A priori convergence theory

In this section, some of the main results of a priory convergence theory for problems in one parameter ($P = 1$) derived in [37, 38, 55] will very briefly be presented. Details about the derivation of the results and proofs can be found in these references.

First, we define the parameter domain $D = [\mu_{\min}, \mu_{\max}] = [1/\sqrt{\mu_r}, \sqrt{\mu_r}]$. We introduce then the non-hierarchical Lagrange equi-ln spaces $W_N^{N_t\,ln}$, $1 \leq N \leq N_{\max}$, given by

$$W_N^{N_t\,ln} = \text{span}\{u^{N_t}(\mu_N^n),\, 1 \leq n \leq N\}, \tag{2.35}$$

for the parameter points given by

$$\mu_N^n = \mu_{\min} \exp\left(\frac{n-1}{N-1} \ln\left(\frac{\mu_{\max}}{\mu_{\min}}\right)\right), \quad 1 \leq n \leq N \leq N_{\max}. \tag{2.36}$$

These spaces contain certain optimality properties as the a priory theory suggests. We obtain the following result:

For any $N \geq N_{\text{crit}}$ and $\forall \mu \in D$,

$$\frac{\|u^{N_t}(\mu) - u_N^{N_t\,ln}(\mu)\|_\mu}{\|u^{N_t}(\mu)\|_\mu} \leq \exp\left(-\frac{N-1}{N_{\text{crit}}-1}\right), \tag{2.37}$$

where

$$N_{\text{crit}} = 1 + [2e \ln \mu_r]_+. \tag{2.38}$$

Here, $[\cdot]_+$ returns the smallest integer greater than or equal to its real argument. This result is not particularly sharp as shown by numerical tests [55], but it leads to several important conclusions [68].

First, the interpretation of RB approximation in a "parameter domain" analogue to FE approximation in the "physical domain" has also a quantitative relevance. Second, while FE convergence relies on spatial regularity, RB convergence is based on smoothness in parameter and discontinuities in space may be allowed. Third, the RB convergence rate upper bound (2.37) does not depend on N_t. The actual convergence rate however does depend on the underlying FE approximation space, but this dependence vanishes as N_t increases for any fixed N. The next conclusion is that the RB convergence rate depends only relatively weakly on the extent of the parameter domain as the exponent in the convergence rate decreases only logarithmically with μ_r. Last, we can confirm that the RB approximation can converge very quickly (exponentially).

For higher parameter dimensions $P > 1$, there is unfortunately no closed a priory convergence theory. Numerical examinations show however that there is also a very rapid convergence in this case (the convergence rate with N appears to depend only weakly on P) and that the RB method can in fact treat problems with "many" parameters.

2.4 A Posteriori Error Estimation

In this section, one of the most important ingredients of this reduced basis methodology will be presented - the a posteriori error estimators. We will first illuminate the special role of the a posteriori error estimation in the reduced basis context in detail. After the introduction of some preliminaries, a posteriori error bounds for affinely parametrized elliptic coercive PDEs will be presented, followed by an offline-online procedure decomposition to guarantee an efficient computation. The last part of this section deals with an application of the error bounds particularly important for (constrained) optimization problems: upper and lower bounds for the "truth" outputs of the underlying FE problem.

2.4.1 Role and Requirements

Effective a posteriori error bounds for the outputs of interest are of great importance for the reduced basis framework. They are crucial for both efficiency and reliability of the reduced basis

approximation. Concerning efficiency, both the offline and online computational procedure benefit from the application of error bounds. As already mentioned before, the greedy algorithm can use a significantly larger training sample with (at the same time) considerably decreased computational costs if a posteriori error estimators are used instead of the real error. This leads to a better accuracy of the reduced basis approximation which can be achieved with a smaller number N of basis functions - this means that we have in turn computational savings in the online stage. Another possibility to save online computational time is to use the error bounds directly in the online stage to find the smallest RB dimension N that allows us to achieve a certain prescribed accuracy. To sum up, the a posteriori error bounds are an essential tool to control the error and hence to minimize the computational costs.

The second keyword - reliability - has a more direct connection to error bounds. We can only have confidence in a method to approximate the solution of a parametrized PDE, if we have some information about the error in our solution and outputs. As the offline sampling procedures only work with a finite training set of parameter points, the error for large parts of our parameter domain D remains uncharacterized. By the help of an efficient a posteriori error bound, we can make up for this error quantification for each new parameter value $\boldsymbol{\mu}$ in the online stage and thus can make sure that constraints are satisfied, feasibility (and safety/failure) conditions are verified and prognoses are valid (in each case not only for the RB approximation but for the "truth" FE solution). That means that we do not loose any confidence in the solution compared to the underlying FE solution while exploiting the rapid predictive power of the RB approximation.

In addition, the pre-asymptotic and essentially ad hoc or empirical nature of reduced basis discretization together with the fact that the RB basis functions can not be directly related to any spatial or temporal scales (which makes physical intuition of little value), and the special needs of deployed real-time systems virtually demand rigorous a posteriori error bounds.

All these reasons for a posteriori error estimation also put several particular requirements on our error bounds. We already mentioned them in section 2.3.5: First, the error bounds must be *rigorous*, that means valid (greater or equal the true error) for all N and all parameter values in the parameter domain D. This is crucial for reliability, as non-rigorous error "indicators" would not suffice for this goal. The second quality we want our error bounds to possess is *sharpness*. If the error bound is too pessimistic or conservative, inefficient approximations with N too large and suboptimal engineering results with unnecessarily high security margins would be the result. We note that even though it is true that 100% error in the output $s_N^{N_t}(\boldsymbol{\mu})$ would be absolutely intolerable, such an error (or even larger errors) in the error estimator is tolerable and not at all useless. But nevertheless, the design should be determined by the true output and not by the (pessimistic) output error. The third requirement is of course *efficiency* - the online operation count and storage to compute the error bounds must be independent of N_t and should be comparable to the cost of the RB output prediction. We should not forget that these error bounds are constructed to measure the error between the reduced basis and underlying "truth" finite element approximation and not relative to the "exact" solution, therefore the FE approximation should be chosen conservatively.

2.4.2 Preliminaries

We first introduce two basic ingredients of our error bounds: The error residual relationship and coercivity lower bounds. The residual $r(v;\boldsymbol{\mu}) \in (X^{N_t})'$ (the dual space to X^{N_t}) is defined as

$$r(v;\boldsymbol{\mu}) \equiv f(v;\boldsymbol{\mu}) - a(u_N^{N_t}(\boldsymbol{\mu}), v;\boldsymbol{\mu}), \quad \forall v \in X^{N_t}. \tag{2.39}$$

Together with

$$f(v;\boldsymbol{\mu}) = a(u^{N_t}, v;\boldsymbol{\mu}), \quad \forall v \in X^{N_t}, \tag{2.40}$$

and the bilinearity of a, we can establish the error residual relationship for the error $e(\boldsymbol{\mu}) \equiv u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}) \in X^{N_t}$:

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = r(v; \boldsymbol{\mu}), \quad \forall v \in X^{N_t}. \tag{2.41}$$

We will also introduce the Riesz representation of $r(v; \boldsymbol{\mu})$: $\hat{e}(\boldsymbol{\mu}) \in X^{N_t}$ [55] satisfies

$$(\hat{e}(\boldsymbol{\mu}), v)_X = r(v; \boldsymbol{\mu}), \quad \forall v \in X^{N_t}. \tag{2.42}$$

This allows us to write (2.39) as

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = (\hat{e}(\boldsymbol{\mu}), v)_X, \quad \forall v \in X^{N_t}, \tag{2.43}$$

and it follows that the dual norm of the residual can be evaluated through the Riesz representation:

$$\|r(\cdot, \boldsymbol{\mu})\|_{(X^{N_t})'} \equiv \sup_{v \in X^{N_t}} \frac{r(v; \boldsymbol{\mu})}{\|v\|_X} = \|\hat{e}(\boldsymbol{\mu})\|_X. \tag{2.44}$$

This shall prove to be important for the offline-online decomposition procedures.

As a second ingredient, we need a positive lower bound $\alpha_{LB}^{N_t}(\boldsymbol{\mu})$ for the coercivity constant $\alpha^{N_t}(\boldsymbol{\mu})$:

$$0 \leq \alpha_{LB}^{N_t}(\boldsymbol{\mu}) \leq \alpha^{N_t}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in D; \tag{2.45}$$

The online computational time to evaluate $\boldsymbol{\mu} \to \alpha_{LB}^{N_t}(\boldsymbol{\mu})$ has to be independent of N_t in order to fulfill the efficiency requirements on the error bounds articulated before.

2.4.3 Error Bounds

We define our error estimator for the energy norm as:

$$\Delta_N^{\text{en}}(\boldsymbol{\mu}) \equiv \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{\sqrt{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}}. \tag{2.46}$$

An equivalent estimator for the output error is defined as

$$\Delta_N^{s}(\boldsymbol{\mu}) \equiv \frac{\|\hat{e}(\boldsymbol{\mu})\|_X^2}{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}. \tag{2.47}$$

We also introduce the effectivities associated to these error estimators in the energy norm, analogue to (2.27), that help us to quantify *rigor* and *sharpness* and thus the quality of the proposed estimator:

$$\eta_N^{\text{en}}(\boldsymbol{\mu}) \equiv \frac{\Delta_N^{\text{en}}}{\|u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}}, \tag{2.48}$$

and

$$\eta_N^{s}(\boldsymbol{\mu}) \equiv \frac{\Delta_N^{s}(\boldsymbol{\mu})}{s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu})}. \tag{2.49}$$

As already stated in (2.3.5), the effectivities should be as close as possible to unity for sharpness, and ≥ 1 for rigor.

We will now derive some results that state that the error bounds introduced above indeed fulfill the requirements of rigor and sharpness [55]. If we choose $v = e(\boldsymbol{\mu})$ in (2.43) it follows with the

Cauchy-Schwartz inequality and the definition of the coercivity constant (2.8)

$$\begin{aligned}\alpha^{N_t}(\boldsymbol{\mu})\|e(\boldsymbol{\mu})\|_X^2 &\leq a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ &\equiv \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2 \\ &= (\hat{e}(\boldsymbol{\mu}), e(\boldsymbol{\mu}))_X \\ &\leq \|\hat{e}(\boldsymbol{\mu})\|_X \|e(\boldsymbol{\mu})\|_X.\end{aligned} \tag{2.50}$$

From the inequalities in (2.50) together with the definition of the effectivity (2.46), we can conclude that

$$\eta_N^{\text{en}}(\boldsymbol{\mu}) \geq 1, \tag{2.51}$$

that means that our energy error bound is indeed a rigorous upper bound for the error measured in the energy norm - feasibility and safety are guaranteed. A similar procedure, but now with $v = \hat{e}(\boldsymbol{\mu})$ in (2.44) gives us (again with the Cauchy-Schwartz inequality and this time with the definition of the continuity constant (2.9)):

$$\begin{aligned}\|\hat{e}(\boldsymbol{\mu})\|_X^2 &= a(e(\boldsymbol{\mu}), \hat{e}(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ &\leq \|\hat{e}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \\ &\leq (\gamma^e(\boldsymbol{\mu}))^{\frac{1}{2}} \|\hat{e}(\boldsymbol{\mu})\|_X \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}.\end{aligned} \tag{2.52}$$

Now the definition of the effectivity (2.48) together with the inequalities (2.52) lead to the following result:

$$\begin{aligned}\Delta_N^{\text{en}}(\boldsymbol{\mu}) &= (\alpha_{LB}^{N_t}(\boldsymbol{\mu}))^{-\frac{1}{2}} \|\hat{e}(\boldsymbol{\mu})\|_X \\ &\leq (\gamma^e(\boldsymbol{\mu}))^{\frac{1}{2}} (\alpha_{LB}^{N_t}(\boldsymbol{\mu}))^{-\frac{1}{2}} \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \\ \Rightarrow \eta_N^{\text{en}}(\boldsymbol{\mu}) &\leq \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}}\end{aligned} \tag{2.53}$$

This result states that the energy error bound overestimates the true error by at most $\sqrt{(\gamma^e(\boldsymbol{\mu})/\alpha_{LB}^{N_t}(\boldsymbol{\mu}))}$, independent of N, and hence is stable with respect to RB refinement. Stability with respect to FE refinements can be achieved if we can find a lower bound for the coercivity constant $\alpha_{LB}^{N_t}(\boldsymbol{\mu})$ which depends only on $\boldsymbol{\mu}$, or if $\alpha^{N_t}(\boldsymbol{\mu})/\alpha_{LB}^{N_t}(\boldsymbol{\mu})$ is bounded by a constant for most $\boldsymbol{\mu} \in D$. The effectivity is then bounded by [68]

$$\eta_N^{\text{en}} \leq \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}} \leq \sqrt{\frac{\alpha^{N_t}(\boldsymbol{\mu})}{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}} \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha^e(\boldsymbol{\mu})}}, \quad \forall \boldsymbol{\mu} \in D. \tag{2.54}$$

We will present the construction of such a lower bound for the coercivity constant in section 2.5.

Similar results can be proven for the output error bound if we use equation (2.31), which states that $s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu}) = \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2$ and $\Delta_N^s(\boldsymbol{\mu}) = (\Delta_N^{\text{en}}(\boldsymbol{\mu}))^2$:

$$\eta_N^s(\boldsymbol{\mu}) = \frac{(\Delta_N^{\text{en}}(\boldsymbol{\mu}))^2}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2} = (\eta_N^{\text{en}}(\boldsymbol{\mu}))^2 \tag{2.55}$$

To sum up, for any $N = 1, \ldots, N_{\max}$, the effectivities satisfy

$$1 \le \eta_N^{\text{en}}(\boldsymbol{\mu}) \le \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}}, \quad \forall \boldsymbol{\mu} \in D, \tag{2.56}$$

$$1 \le \eta_N^{s}(\boldsymbol{\mu}) \le \frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{N_t}(\boldsymbol{\mu})}, \qquad \forall \boldsymbol{\mu} \in D. \tag{2.57}$$

2.4.4 Offline-Online Procedure

The error bounds developed in the previous section are only useful if they allow for an efficient offline-online computational procedure that leads to an online complexity independent of N_t. The offline-online decomposition presented in the following is mainly based on the dual norm of the residual. With the affine decomposition of $u_N^{N_t}(\boldsymbol{\mu})$ (2.11) and the expansion of $u_N^{N_t}(\boldsymbol{\mu})$ in the N basis functions (2.19), the residual can be expressed as

$$\begin{aligned} r(v;\boldsymbol{\mu}) &= f(v) - a(u_N^{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) \\ &= f(v) - \sum_{n=1}^{N} u_{N\,n}^{N_t}(\boldsymbol{\mu}) \sum_{q=1}^{Q_a} \Theta^q(\boldsymbol{\mu}) a^q(\zeta_n^{N_t}, v) \end{aligned} \tag{2.58}$$

Together with (2.42) and linear superposition, this gives us

$$(\hat{e}(\boldsymbol{\mu}), v)_X = f(v) - \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \Theta^q(\boldsymbol{\mu}) u_{N\,n}^{N_t}(\boldsymbol{\mu}) a^q(\zeta_n^{N_t}, v). \tag{2.59}$$

It then follows that we may write $\hat{e}(\boldsymbol{\mu}) \in X^{N_t}$ as

$$\hat{e}(\boldsymbol{\mu}) = \mathcal{F} + \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \Theta^q(\boldsymbol{\mu}) u_{N\,n}^{N_t}(\boldsymbol{\mu}) \mathcal{A}_n^q, \tag{2.60}$$

where $\mathcal{F} \in X^{N_t}$ and $\mathcal{A}_n^q \in X^{N_t}$ (called FE "pseudo"-solutions) satisfy

$$(\mathcal{F}, v)_X = f(v), \quad \forall v \in X^{N_t}, \tag{2.61}$$

$$(\mathcal{A}_n^q, v)_X = -a^q(\zeta_n^{N_t}, v), \quad \forall v \in X^{N_t},\ n \in \mathbb{N},\ 1 \le q \le Q_a. \tag{2.62}$$

We note that (2.61) and (2.62) are simple parameter-independent Poisson or Poisson-like problems and thus can be solved once in the offline stage. It then follows the full decomposition of $\|\hat{e}(\boldsymbol{\mu})\|_X$:

$$\begin{aligned} \|\hat{e}(\boldsymbol{\mu})\|_X^2 &= \left(\mathcal{F} + \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \Theta^q(\boldsymbol{\mu}) u_{N\,n}^{N_t}(\boldsymbol{\mu}) \mathcal{A}_n^q,\ \mathcal{F} + \sum_{q'=1}^{Q_a} \sum_{n'=1}^{N} \Theta^{q'}(\boldsymbol{\mu}) u_{N\,n'}^{N_t}(\boldsymbol{\mu}) \mathcal{A}_{n'}^{q'} \right)_X \\ &= (\mathcal{F}, \mathcal{F})_X + \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \Theta^q(\boldsymbol{\mu}) u_{N\,n}^{N_t}(\boldsymbol{\mu}) \left\{ 2(\mathcal{F}, \mathcal{A}_n^q)_X + \sum_{q'=1}^{Q_a} \sum_{n'=1}^{N} \Theta^{q'}(\boldsymbol{\mu}) u_{N\,n'}^{N_t}(\boldsymbol{\mu}) (\mathcal{A}_n^q, \mathcal{A}_{n'}^{q'})_X \right\}. \end{aligned} \tag{2.63}$$

This expression can be related to the requisite dual norm of the residual through (2.44). It is the sum of products of parameter-dependent known functions and parameter independent inner products, formed of more complicated but precomputable quantities. The offline-online decomposition is thus clear.

In the offline stage we first solve (2.61), (2.62) for the parameter-independent FE "pseudo"-solutions $\mathcal{F}$ and $\mathcal{A}_n^q$, $1 \le n \le N_{\max}$, $1 \le q \le Q_a$ and form/store the parameter-independent inner products $(\mathcal{F},\mathcal{F})_X$, $(\mathcal{F},\mathcal{A}_n^q)_X$, $(\mathcal{A}_n^q,\mathcal{A}_n^q)_X$, $1 \le n \le N_{\max}$, $1 \le q \le Q_a$. The offline operation count depends then on $N_{\max}$, Q_a and N_t.

In the online stage - performed for each new value of $\boldsymbol{\mu}$ - we simply evaluate the sum (2.63) in terms of the $\Theta^q(\boldsymbol{\mu})$, $1 \le q \le Q_a$ and $u_{N\,n}^{N_t}(\boldsymbol{\mu})$, $1 \le n \le N$ (already computed for the output evaluation) and the precalculated and stored (parameter-independent) $(\cdot,\cdot)_X$ inner products. The online operation count, and hence the marginal and asymptotic average cost, is only $\mathcal{O}(Q_a^2 N^2)$, and thus the crucial point - the independence of N_t - is again achieved. We further note that, unless Q_a is quite large, the online cost associated with the calculation of the residual dual norm and the online cost associated with the calculation of $s_N^{N_t}(\boldsymbol{\mu})$ are comparable. Again, the hierarchical properties of our reduced basis approximation spaces allows us to simply extract the necessary quantities for any $N \in \{1,\dots,N_{\max}\}$ from the corresponding quantities for $N = N_{\max}$.

2.4.5 Upper and Lower Bounds for the Outputs

The output error estimators introduced in the previous sections for the compliant case can serve us to compute reliable upper and lower bounds $s_N^{N_t\,+}(\boldsymbol{\mu})$ and $s_N^{N_t\,-}(\boldsymbol{\mu})$ for the "truth" output $s^{N_t}(\boldsymbol{\mu})$ for each new value for $\boldsymbol{\mu}$ based on the reduced basis output $s_N^{N_t}(\boldsymbol{\mu})$, such that

$$s_N^{N_t\,+}(\boldsymbol{\mu}) \ge s^{N_t}(\boldsymbol{\mu}) \ge s_N^{N_t\,-}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in D,\, \forall N \in [1,\dots,N_{\max}]. \tag{2.64}$$

We establish these limits as

$$s_N^{N_t\,-}(\boldsymbol{\mu}) = s_N^{N_t}(\boldsymbol{\mu}), \tag{2.65}$$

$$s_N^{N_t\,+}(\boldsymbol{\mu}) = s_N^{N_t}(\boldsymbol{\mu}) + \Delta_N^s(\boldsymbol{\mu}). \tag{2.66}$$

To demonstrate that equation (2.65) is indeed valid, we note that in the compliant case

$$\begin{aligned} s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu}) &= f(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu})) \\ &= a(u^{N_t}(\boldsymbol{\mu}), u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ &= a(u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}), u^{N_t}(\boldsymbol{\mu}) - u_N^{N_t}(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ &\ge 0. \end{aligned} \tag{2.67}$$

This results of the symmetry of a, Galerkin orthogonality (2.33) and coercivity. This important result (which has also been stated in section 2.3.6) confirms that our reduced basis approximation is a lower bound for the FE solution.

The validity of the upper bound (2.66) follows directly from (2.57):

$$\begin{aligned} \eta_N^s(\boldsymbol{\mu}) \ge 1 \quad &\Rightarrow \quad \Delta_N^s(\boldsymbol{\mu}) \ge s^{N_t}(\boldsymbol{\mu}) - s_N^{N_t}(\boldsymbol{\mu}) \\ &\Rightarrow \quad s_N^{N_t\,+}(\boldsymbol{\mu}) = s_N^{N_t}(\boldsymbol{\mu}) + \Delta_N^s(\boldsymbol{\mu}) \ge s^{N_t}(\boldsymbol{\mu}). \end{aligned} \tag{2.68}$$

These upper and lower bounds for the underlying "truth" FE output play an important role for example in optimization problems. They assure that possible constraints are not only met for the reduced basis output, but also for the "truth" output. Also in many other applications this property has a great importance.

2.5 Coercivity Lower Bounds

In this chapter, we will concentrate on the efficient computation of lower bounds for the coercivity constant (2.8). They are an essential ingredient of the a posteriori error bounds developed in the previous chapter. The discrete coercivity constant (2.8) is a generalized eigenvalue problem [55], and there exist a number of classical techniques for the numerical approximation of eigenvalues. However, many of them are not suited for the particular needs in the reduced basis context [68]. Therefore, we will present here the Successive Constraint Method (SCM) described in [24]. This algorithm has been developed for the special requirements of the reduced basis method and thus features an efficient offline-online strategy which makes the online calculation complexity independent of N_t - a fundamental requisite. The SCM is more efficient and general than other, earlier approaches [41, 59, 72]. In addition, it is easier to implement.

We will consider two different classes of affine problems. At first, we will present the (easy to compute) coercivity lower bounds for the special case of parametrically coercive problems. This presentation will be followed by the more general case, where the problems are parameter-dependent and coercive, but do not fulfill the requirements for parametrical coercivity.

2.5.1 Parametrically Coercive Problems

We shall now assume that our bilinear form a is not only affine in the parameter, symmetric and coercive, but also parametrically coercive. This means that the affine decomposition of a fulfills

$$a(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta^q(\boldsymbol{\mu}) a^q(w, v), \quad \forall w, v \in X^e, \tag{2.69}$$

where

$$\Theta^q(\boldsymbol{\mu}) > 0, \quad \forall \boldsymbol{\mu} \in D,\, 1 \leq q \leq Q_a \tag{2.70}$$

and

$$a^q(w, w) \geq 0, \quad \forall w \in X^e,\, 1 \leq q \leq Q_a. \tag{2.71}$$

A positive lower bound for the coercivity constant, $0 \leq \alpha_{LB}^{N_t}(\boldsymbol{\mu}) \leq \alpha^{N_t}(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in D$, is then developed relatively easily. Indeed, the hypothesis of parametrical coercivity allows us the explicit construction and calculation of a lower bound. We first define the easily evaluated functions

$$\Theta_a^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) = \min_{q \in [1,\ldots,Q_a]} \frac{\Theta^q(\boldsymbol{\mu})}{\Theta^q(\overline{\boldsymbol{\mu}})}. \tag{2.72}$$

We can then show that a suitable lower bound for the coercivity constant is given by

$$0 \leq \Theta_a^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \leq \alpha^{N_t}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in D. \tag{2.73}$$

To prove this result, we note that (from our positivity conditions and choice of norm)

$$\begin{aligned} a(w,w;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_a} \Theta^q(\boldsymbol{\mu}) a^q(w,w) \\ &= \sum_{q=1}^{Q_a} \frac{\Theta^q(\boldsymbol{\mu})}{\Theta^q(\overline{\boldsymbol{\mu}})} \Theta^q(\overline{\boldsymbol{\mu}}) a^q(w,w) \\ &\geq \left(\min_{q\in[1,\ldots,Q_a]} \frac{\Theta^q(\boldsymbol{\mu})}{\Theta^q(\overline{\boldsymbol{\mu}})} \right) a(w,w;\boldsymbol{\mu}) \\ &= \Theta_a^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \|w\|_X^2, \quad \forall w \in X^e, \forall \boldsymbol{\mu} \in D. \end{aligned} \tag{2.74}$$

It follows then

$$\alpha^e(\boldsymbol{\mu}) \equiv \inf_{w\in X^e} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2} \geq \Theta_a^{\min,\overline{\boldsymbol{\mu}}}(\boldsymbol{\mu}) \geq 0, \quad \forall \boldsymbol{\mu} \in D, \tag{2.75}$$

which states the result in (2.73).

We note that similar results can be obtained for non-symmetric bilinear forms if we consider only the symmetric part of the bilinear form [55].

2.5.2 Coercive Problems: The Successive Constraint Method

In the more general case of coercive problems which do not fulfill (2.70), (2.71), the efficient computation of the necessary coercivity lower bounds is a much greater challenge. We still consider a symmetric, continuous and coercive bilinear form a, which is affine in the parameter. However, the following results can be readily extended to non-symmetric operators (see section 2.6.2) and general non-coercive operators (see chapter 7).

We first recall the definition of our FE coercivity constant,

$$\alpha^{N_t}(\boldsymbol{\mu}) = \inf_{w\in X^{N_t}} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2}, \quad \forall \boldsymbol{\mu} \in D. \tag{2.76}$$

We then introduce an objective function $J^{\text{obj}} : D \times \mathbb{R}^{Q_a} \to \mathbb{R}$ given by

$$J^{\text{obj}}(\boldsymbol{\mu}; y) = \sum_{q=1}^{Q_a} \Theta^q(\boldsymbol{\mu}) y_q, \quad \text{with } y = (y_1, \ldots, y_{Q_a}). \tag{2.77}$$

Next, the set $Y \in \mathbb{R}^{Q_a}$ is defined by

$$Y = \left\{ y \in \mathbb{R}^{Q_a} \,\middle|\, \exists w_y \in X^{N_t} \quad \text{s.t. } y_q = \frac{a^q(w_y,w_y)}{\|w_y\|_X^2}, \quad 1 \leq q \leq Q_a \right\}. \tag{2.78}$$

The affine parameter decomposition (2.11) allows us now to express our coercivity constant (2.76) as

$$\alpha^{N_t}(\boldsymbol{\mu}) = \inf_{y\in Y} J^{\text{obj}}(\boldsymbol{\mu}; y). \tag{2.79}$$

The next step is to introduce a "continuity constraint" box

$$B = \prod_{q=1}^{Q_a} \left[\inf_{w\in X^{N_t}} \frac{a^q(w,w)}{\|w\|_X^2}, \sup_{w\in X^{N_t}} \frac{a^q(w,w)}{\|w\|_X^2} \right]; \tag{2.80}$$

which, from our continuity hypothesis, is bounded. The last ingredient is a "coercivity constraint"

sample,

$$C_J = \{\boldsymbol{\mu}^1_{\mathrm{SCM}} \in D, \ldots, \boldsymbol{\mu}^J_{\mathrm{SCM}} \in D\}. \tag{2.81}$$

$C_J^{M,\boldsymbol{\mu}}$ denotes the subset of C_J with the $M\,(\geq 1)$ points closest (in the Euclidian norm) to a given $\boldsymbol{\mu} \in D$. We make the convention $C_J^{M,\boldsymbol{\mu}} = C_J$ if $M > J$.

Lower Bound

Our strategy is to define a set $Y_{LB}(\boldsymbol{\mu}; C_J, M)$ which is relatively easy/cheap to compute and which can be used as *surrogate* for Y in (2.79). For given $C_J \in D$ and $M \in \mathbb{N}$, this set must contain the original set Y, that means

$$Y \subset Y_{LB}(\boldsymbol{\mu}; C_J, M), \quad \forall \boldsymbol{\mu} \in D. \tag{2.82}$$

If we choose

$$Y_{LB}(\boldsymbol{\mu}; C_J, M) \equiv \left\{ y \in \mathbb{R}^{Q_a} \,\middle|\, y \in B, \quad \sum_{q=1}^{Q_a} \Theta^q(\boldsymbol{\mu}') y_q \geq \alpha^{N_t}(\boldsymbol{\mu}'), \quad \forall \boldsymbol{\mu}' \in C_J^{M,\boldsymbol{\mu}} \right\} \tag{2.83}$$

as our "surrogate set", we can prove that (2.82) is indeed fulfilled:

From the definition of Y (2.78) it follows that for any $y \in Y$, $\exists w_y \in X^{N_t}$, such that $y_q = \frac{a^q(w_y, w_y)}{\|w_q\|_X^2}$, $1 \leq q \leq Q_a$. Then, since

$$\inf_{w \in X^{N_t}} \frac{a^q(w,w)}{\|w\|_X^2} \leq \frac{a^q(w_y, w_y)}{\|w_y\|_X^2} \leq \sup_{w \in X^{N_t}} \frac{a^q(w,w)}{\|w\|_X^2}, \tag{2.84}$$

and also

$$\sum_{q=1}^{Q_a} \Theta^q(\boldsymbol{\mu}) \frac{a^q(w_y, w_y)}{\|w_y\|_X^2} = \frac{a(w_y, w_y; \boldsymbol{\mu})}{\|w_y\|_X^2} \geq \alpha^{N_t}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in D, \tag{2.85}$$

we can deduce that every member y of Y is also a member of $Y_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)$, which concludes the proof.

We can now define our lower bound as

$$\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M) = \min_{y \in Y_{LB}(\boldsymbol{\mu}; C_J, M)} J^{\mathrm{obj}}(\boldsymbol{\mu}; y). \tag{2.86}$$

This definition is indeed a correct choice, as we can show with (2.82), that for given $C_J \subset D$ and $M \in \mathbb{N}$

$$\begin{aligned} \alpha_{LB}^{N_t}(\boldsymbol{\mu}) &= \min_{y \in Y_{LB}(\boldsymbol{\mu}; C_J, M)} J^{\mathrm{obj}}(\boldsymbol{\mu}; y) \\ &\leq \min_{y \in Y} J^{\mathrm{obj}}(\boldsymbol{\mu}; y) \\ &= \alpha_{LB}^{N_t}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in D. \end{aligned} \tag{2.87}$$

This means that the necessary requirement for a lower bound,

$$\alpha_{LB}^{N_t}(\boldsymbol{\mu}) \leq \alpha^{N_t}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in D, \tag{2.88}$$

is fulfilled.

The computation of our lower bound (2.86) is in fact a linear optimization problem (or Linear Program (LP)) [68]. It contains Q_a design variables and $2Q_a + M$ inequality constraints. It is important to note that again, this approach allows us to evaluate $\boldsymbol{\mu} \to \alpha_{LB}^{N_t}(\boldsymbol{\mu})$ with a computational

cost independent of N_t, if B and the set $\{\alpha^{N_t}(\boldsymbol{\mu}')|\boldsymbol{\mu}' \in C_J\}$ are given. The (offline) computation of these quantities however is N_t-dependent. We will discuss the offline-online decomposition and computational costs in more detail in section 2.5.2.

Upper Bound

Although it is not directly necessary for our error bounds, we will also compute an upper bound for the coercivity constant. This will serve us for the efficient construction of a good coercivity constraint sample C_J. Similar to the approach for the lower bound, we introduce an "upper bound set" $Y_{UB}(\boldsymbol{\mu}; C_J, M) \in \mathbb{R}^{Q_a}$ as

$$Y_{LB}(\boldsymbol{\mu}; C_J, M) = \left\{ y^*(\boldsymbol{\mu}') | \boldsymbol{\mu}' \in C_J^{M,\boldsymbol{\mu}} \right\}, \tag{2.89}$$

where

$$y^*(\boldsymbol{\mu}) = \arg \inf_{y \in Y} J^{\text{obj}}(\boldsymbol{\mu}; y). \tag{2.90}$$

Our upper bound is then defined by

$$\alpha_{UB}^{N_t}(\boldsymbol{\mu}; C_J, M) = \min_{y \in Y_{LB}(\boldsymbol{\mu}; C_J, M)} J^{\text{obj}}(\boldsymbol{\mu}; y). \tag{2.91}$$

As we can see directly from (2.89), $Y_{UB}(\boldsymbol{\mu}; C_J, M) \subset Y$. It then follows that, for given C_J, $M \in \mathbb{N}$:, $\alpha_{UB}^{N_t}(\boldsymbol{\mu}; C_J, M) \geq \alpha^{N_t}(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in D$. That means our choice for $\alpha_{UB}^{N_t}(\boldsymbol{\mu})$ is indeed suited as upper bound for the coercivity constant.

Again we can state that the operation count for the online evaluation $\boldsymbol{\mu} \to \alpha_{UB}^{N_t}(\boldsymbol{\mu})$ is independent of N_t (if the set $\{y^*(\boldsymbol{\mu}')|\boldsymbol{\mu}' \in C_J\}$ is already given).

Selection of C_J

The selection process for C_J will be based on a greedy algorithm somewhat similar to the greedy selection process for the basis functions described in section 2.3.5. We shall again introduce a "train" sample $\Xi_{\text{train,SCM}} = \{\boldsymbol{\mu}^1_{\text{train,SCM}}, \ldots, \boldsymbol{\mu}^{n_{\text{train,SCM}}}_{\text{train,SCM}}\} \subset D$ of $n_{\text{train,SCM}}$ parameter points and a tolerance $\epsilon_{\text{SCM}} \in [0, 1]$ for the error in the lower bound prediction. We start with $J = 1$ and $C_1 = \{\boldsymbol{\mu}_{\text{SCM}}\}$ chosen arbitrary. The greedy algorithm is then given by

$$\begin{aligned}
&\texttt{While} \quad \alpha^J_{\text{tol}} = \max_{\boldsymbol{\mu} \in \Xi_{\text{train,SCM}}} \left[\frac{\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M) - \alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)}{\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)} \right] > \epsilon_{\text{SCM}} : \\
&\qquad \boldsymbol{\mu}^{J+1}_{\text{SCM}} = \arg \max_{\boldsymbol{\mu} \in \Xi_{\text{train,SCM}}} \left[\frac{\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M) - \alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)}{\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)} \right]; \\
&\qquad C_{J+1} = C_J \cup \boldsymbol{\mu}^{J+1}_{\text{SCM}}; \\
&\qquad J \leftarrow J + 1; \\
&\quad \texttt{end.} \\
&\texttt{Set}\ J_{\max} = J
\end{aligned} \tag{2.92}$$

We choose $\alpha_{UB}(\boldsymbol{\mu}; C_J, M)$ in the denominator of α^J_{tol}, as this value is strictly positive, contrary to $\alpha_{LB}(\boldsymbol{\mu}; C_J, M)$ which may be negative or zero.

Indeed, the strategy is basically the same as in the greedy algorithm in section 2.3.5. In each iteration of the greedy procedure, we add to our "coercivity constraint" sample that point in D for which the current lower bound approximation is least accurate. The true error is thereby replaced by a (computationally cheaper) surrogate which makes it possible to perform a more efficient and

more exhaustive search. Although α_{tol}^J is not decreasing monotonic, we can anticipate that "overall" it is decreasing for increasing J. Indeed, from continuity reasons and since $\alpha_{LB}(\boldsymbol{\mu}; C_J, M) = \alpha_{UB}(\boldsymbol{\mu}; C_J, M)$, $\forall \boldsymbol{\mu} \in C_J$, it follows that our error tolerance will be honored for sufficiently large J. However, if there exist numerical problems for single parameter samples $\boldsymbol{\mu} \in \Xi_{\text{train,SCM}}$, this can lead to divergence of the algorithm although for rest of the train sample $\Xi_{\text{train,SCM}}$ the computation of the coercivity lower bounds is successful. We will provide an example for this phenomenon in the practical part of this work.

Furthermore, it is important to note that our choice of stopping criterion allows us to bound

$$\begin{aligned}
\frac{\alpha^{N_t}(\boldsymbol{\mu})}{\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_{J_{\max}}, M)} &= \frac{\alpha^{N_t}(\boldsymbol{\mu})}{\alpha_{UB}^{N_t}(\boldsymbol{\mu}; C_{J_{\max}}, M)} \cdot \frac{\alpha_{UB}^{N_t}(\boldsymbol{\mu}; C_{J_{\max}}, M)}{\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_{J_{\max}}, M)} \\
&\leq \frac{\alpha^{N_t}(\boldsymbol{\mu})}{\alpha_{UB}^{N_t}(\boldsymbol{\mu}; C_{J_{\max}})} \frac{1}{1-\epsilon_{\text{SCM}}} \\
&\leq \frac{1}{1-\epsilon_{\text{SCM}}}, \qquad \forall \boldsymbol{\mu} \in \Xi_{\text{train,SCM}}. \qquad (2.93)
\end{aligned}$$

This result can be inserted in (2.54) to obtain the upper bounds for the effectivities which are now independent of N and N_t. Usually, we may set $\epsilon_{\text{SCM}} = 0.75$, which is rather crude. Nevertheless this choice has relatively little detrimental effect on our error bounds.

Offline-Online Procedure

Finally, we precise the offline-online decomposition for the computation of the coercivity lower bounds and give an overview of the operation counts involved.

In the offline stage, we first have to build B and the set $\{\alpha^{N_t}(\boldsymbol{\mu}')|\boldsymbol{\mu}' \in C_{J_{\max}}\}$. This results in $2Q_a$ resp. $J_{\max}$ eigenproblems over X^{N_t}. Second, we have to subsequently form the set $\{y^*(\boldsymbol{\mu}')|\, \boldsymbol{\mu}' \in C_{J_{\max}}\}$ ($J_{\max}Q_a$ inner products over X^{N_t}). The last offline step is the solution of the $n_{\text{train,SCM}}J_{\max}$ linear optimization problems of "size" $2Q_a + M$ to perform the "arg max". The offline computational cost thus roughly scales as $\mathcal{O}(N_t \cdot (2Q_a + J_{\max})) + \mathcal{O}(N_t Q_a J_{\max}) + \mathcal{O}(n_{\text{train,SCM}} J_{\max} Q_a M)$. As already mentioned before, the offline computational costs of course depend on N_t, but N_t and $n_{\text{train,SCM}}$ do not occur as a product in any of the terms - there is no term $\mathcal{O}(N_t n_{\text{train,SCM}})$. This means that we can choose both N_t and $n_{\text{train,SCM}}$ very large.

For each evaluation $\boldsymbol{\mu} \to \alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)$ in the online stage, we first perform a sort of the $J_{\max}$ points in $C_{J_{\max}}$ to determine the set $C_{J_{\max}}^{M,\boldsymbol{\mu}}$. The operation count here is at most $\mathcal{O}(MJ_{\max})$. Then we must perform the $(M+1)Q_a$ evaluations $\boldsymbol{\mu}' \to \theta^q(\boldsymbol{\mu}')$, $1 \leq q \leq Q_a$, which results in an operation count of $\mathcal{O}((M+1)Q_a)$. The last step is the extraction of the selected M members of the pre-computed set $\{\alpha^{N_t}(\boldsymbol{\mu}')|\boldsymbol{\mu}' \in C_J\}$ and the solution of the resulting linear optimization problem to obtain $\alpha_{LB}^{N_t}(\boldsymbol{\mu}; C_J, M)$. The fundamental point is again that the online evaluation count is independent of N_t and the rapid evaluation of the error bounds is supported by the SCM procedure.

2.6 Extensions

In the last sections we presented a complete reduced basis methodology for affinely parametrized elliptic coercive partial differential equations in the compliant case, that means for $s = f$. This methodology will be extended in section 2.6.1 to noncompliant problems where $s \neq f$, using a primal-dual approximation framework. Nonsymmetric problems are generally not compliant, even if $s = f$ holds. The (relatively small) additional modifications necessary for the asymmetric case are then described in section 2.6.2.

2.6.1 Noncompliant Problems

We first recall the general non-compliant problem: given $\boldsymbol{\mu} \in D$, solve

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \quad \forall v \in X^e, \tag{2.94}$$

and then evaluate

$$s(\boldsymbol{\mu}) = l(u^e(\boldsymbol{\mu})), \tag{2.95}$$

where a is a coercive, continuous and affine bilinear form. It is no longer required that $l = f$, but l and f both have to be bounded linear functionals. The new aspect is that we also consider the dual problem associated to l: find $\psi^e(\boldsymbol{\mu}) \in X^e$ (the "adjoint" our "dual" field), s.t.

$$a(v, \psi^e(\boldsymbol{\mu}); \boldsymbol{\mu}) = -l(v), \quad \forall v \in X^e; \tag{2.96}$$

The corresponding FE approximation of (2.94) and (2.96) is then given by

$$a(u_{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \quad \forall v \in X^{N_t} \tag{2.97}$$

and

$$a(v, \psi^{N_t}(\boldsymbol{\mu}); \boldsymbol{\mu}) = -l(v), \quad \forall v \in X^{N_t} \tag{2.98}$$

respectively.

Primal-Dual Framework

We now turn to the reduced basis approximation of this problem. The first step is to introduce a general Petrov-Galerkin approximation of the primal problem which will give us the motivation for the full primal-dual framework. For this purpose, we need two reduced basis approximation subspaces, the "trial" space $\tilde{X}_1^{\mathrm{pr}} \subset X^{N_t}$ and the "test" space $\tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu}) \subset X^{N_t}$. The second subspace can be parameter dependent. In addition, we introduce a given function $\Phi : D \to X^{N_t}$. The particular properties of the approximation spaces and Φ are not important yet. The general Petrov-Galerkin approximation is then defined as follows:

Given $\boldsymbol{\mu} \in D$, find $\tilde{u}^{N_t}(\boldsymbol{\mu}) \in \tilde{X}_1^{\mathrm{pr}}$ such that

$$a(\tilde{u}^{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \quad \forall v \in \tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu}), \tag{2.99}$$

and evaluate the output as

$$\tilde{s}^{N_t}(\boldsymbol{\mu}) = l(\tilde{u}^{N_t}(\boldsymbol{\mu})) - \tilde{r}^{\mathrm{pr}}(\Phi(\boldsymbol{\mu}); \boldsymbol{\mu}), \tag{2.100}$$

with the primal residual $\tilde{r}^{\mathrm{pr}}(v; \boldsymbol{\mu})$ defined equivalent to (2.39):

$$\tilde{r}^{\mathrm{pr}}(v; \boldsymbol{\mu}) \equiv f(v) - a(\tilde{u}^{N_t}(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \quad \forall v \in X^e. \tag{2.101}$$

Our RB approximations to the field $u^{N_t}(\boldsymbol{\mu})$ and output $s^{N_t}(\boldsymbol{\mu})$ are denoted with $\tilde{u}^{N_t}(\boldsymbol{\mu})$ and $\tilde{s}^{N_t}(\boldsymbol{\mu})$, respectively.

Although in the coercive case standard Galerkin RB approximation is sufficient, $\tilde{X}_1^{\mathrm{pr}} = \tilde{X}_2^{\mathrm{pr}}$, we want to illustrate the different roles of the trial and test space in the full Petrov-Galerkin context. The discrete coercivity, inf-sup and continuity constants in this setup are given as

$$\tilde{\alpha}(\boldsymbol{\mu}) = \inf_{w \in \tilde{X}_1^{\mathrm{pr}}} \frac{a(w, w; \boldsymbol{\mu})}{\|w\|_X^2}, \tag{2.102}$$

$$\tilde{\beta}(\boldsymbol{\mu}) = \inf_{w \in \tilde{X}_1^{\mathrm{pr}}} \sup_{v \in \tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu})} \frac{a(w, v; \boldsymbol{\mu})}{\|w\|_X^2 \|v\|_X^2} \tag{2.103}$$

and

$$\tilde{\gamma}(\boldsymbol{\mu}) = \sup_{w \in \tilde{X}_1^{\mathrm{pr}}} \sup_{v \in \tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu})} \frac{a(w, v; \boldsymbol{\mu})}{\|w\|_X^2 \|v\|_X^2}. \tag{2.104}$$

We can then obtain a result similar to the standard Galerkin optimality results (2.29) to (2.32) for $\tilde{u}^{N_t}(\boldsymbol{\mu})$ and $\tilde{s}^{N_t}(\boldsymbol{\mu})$:

$$\|u^{N_t}(\boldsymbol{\mu}) - \tilde{u}^{N_t}(\boldsymbol{\mu})\|_X \leq \left(1 + \frac{\gamma^e(\boldsymbol{\mu})}{\tilde{\beta}(\boldsymbol{\mu})}\right) \inf_{\tilde{w} \in X_1^{\mathrm{pr}}} \|u^{N_t}(\boldsymbol{\mu}) - \tilde{w}\|_X \tag{2.105}$$

and

$$|s^{N_t}(\boldsymbol{\mu}) - \tilde{s}^{N_t}(\boldsymbol{\mu})| \leq \gamma^e(\boldsymbol{\mu}) \|u^{N_t}(\boldsymbol{\mu}) - \tilde{u}^{N_t}(\boldsymbol{\mu})\|_X \inf_{\tilde{v} \in X_2^{\mathrm{pr}}(\boldsymbol{\mu})} \|\Psi^{N_t}(\boldsymbol{\mu}) - \Phi(\boldsymbol{\mu}) - \tilde{v}\|_X. \tag{2.106}$$

In this context, $\Phi : D \to X^{N_t}$ can be any given function.

The role of the test space $\tilde{X}_2^{\mathrm{pr}}$ is thus twofold. First, it determines the stability of the approximation through $\tilde{\beta}(\boldsymbol{\mu})$, both concerning algebraic conditioning and convergence. As already mentioned, in the coercive case the standard Galerkin choice of equivalent trial and test space is suited to ensure stability, for noncoercive problems however, $\tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu})$ can play an important role to establish stability. Second, we can conclude from equation (2.106) that the choice of $\tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu})$ also has a direct influence on the accuracy of the output approximation. If we choose $\tilde{X}_2^{\mathrm{pr}}(\boldsymbol{\mu})$ such that its members can approximate $\Psi^{N_t}(\boldsymbol{\mu})$, that means the adjoint solution, very well, then the second term in (2.106) will be very small and consequently the output error will be small. This can improve the accuracy of the output prediction for a given trial space $\tilde{X}_1^{\mathrm{pr}}$.

We note that we still have a degree of freedom left - $\Phi(\boldsymbol{\mu})$ is still undefined. With the choice of the test space $\tilde{X}_2^{\mathrm{pr}}$, we can either aim at stability or at accuracy - it is (in most of the cases) not possible to exploit the qualities of the test space both for stability considerations and for the improvement of accuracy. The introduction of $\Phi(\boldsymbol{\mu})$ in equation (2.100) allows us to resolve this dilemma, as also the choice of $\Phi(\boldsymbol{\mu})$ has an influence on the accuracy of the prediction. We can thus choose $\tilde{X}_2^{\mathrm{pr}}$ in a way that controls stability, and select $\Phi(\boldsymbol{\mu})$ to control the second term in (2.106).

The particular choice of $\Phi(\boldsymbol{\mu})$ will typically be a Galerkin (or Petrov-Galerkin) approximation of the dual problem: Given $\boldsymbol{\mu} \in D$, find $\tilde{\Psi}^{N_t}(\boldsymbol{\mu}) = \Phi(\boldsymbol{\mu}) \in \tilde{X}_1^{\mathrm{du}}$ such that

$$a(v, \tilde{\Psi}^{N_t}(\boldsymbol{\mu}); \boldsymbol{\mu}) = -l(v), \quad \forall v \in \tilde{X}_2^{\mathrm{du}}(\boldsymbol{\mu}), \tag{2.107}$$

Here, the dual trial space $\tilde{X}_1^{\mathrm{du}}$ shall provide a good approximation of $\Psi^{N_t}(\boldsymbol{\mu})$ while the dual test space $\tilde{X}_2^{\mathrm{du}}(\boldsymbol{\mu})$ is chosen to control the stability of the discrete dual problem. We note here that the primal and dual approximation spaces can differ considerably in the RB context due to the "ad hoc" character of the approximation. As for the primal problem the standard Galerkin approach with $\tilde{X}_1^{\mathrm{du}} = \tilde{X}_2^{\mathrm{du}}$ is sufficient in the coercive case.

RB Methodology in the Noncompliant Case

For the application of this approach in our RB methodology, we choose (greedy) Lagrange RB spaces as primal and dual approximation spaces and follow the Galerkin recipe (stability guaranteed): $\tilde{X}_1^{\mathrm{pr}} = \tilde{X}_2^{\mathrm{pr}} = W_{N^{\mathrm{pr}}}^{\mathrm{pr}}$ and $\tilde{X}_1^{\mathrm{du}} = \tilde{X}_2^{\mathrm{du}} = W_{N^{\mathrm{du}}}^{\mathrm{du}}$. This means the primal RB approximation spaces are built by the span of N^{pr} snapshots of the primal solution $u^{N_t}(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in D$, while the dual RB approximation spaces are built by the span of N^{du} snapshots of the dual solution $\Psi^{N_t}(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in$

D. In the primal-dual framework, we first have to compute both the RB primal and dual field solutions, which are denoted by $u^{N_t}_{N^{\mathrm{pr}}}(\boldsymbol{\mu})$ and $\Psi^{N_t}_{N^{\mathrm{du}}}(\boldsymbol{\mu})$, respectively. In the following, we will use the convention that $u^{N_t}_0(\boldsymbol{\mu}) = 0$ if $N^{\mathrm{pr}} = 0$ and $\Psi^{N_t}_0(\boldsymbol{\mu}) = 0$ in the case $N^{\mathrm{du}} = 0$. In the second step, we can use the primal and dual field solutions to evaluate the associated RB output approximation for $\Phi(\boldsymbol{\mu}) = \Psi^{N_t}_{N^{\mathrm{du}}}(\boldsymbol{\mu})$:

$$s^{N_t}_{N^{\mathrm{pr}},N^{\mathrm{du}}}(\boldsymbol{\mu}) = l(u^{N_t}_{N^{\mathrm{pr}}}(\boldsymbol{\mu})) - r^{\mathrm{pr}}(\Psi^{N_t}_{N^{\mathrm{du}}}(\boldsymbol{\mu});\boldsymbol{\mu}). \tag{2.108}$$

To give an estimation of the computational savings we can obtain by the use of the primal-dual framework, we introduce a monotonically decreasing "convergence" function $g_{\mathrm{err}} : \mathbb{N}_0 \to \mathbb{R}$ with associated inverse g^{-1}_{err} such that $g^{-1}_{\mathrm{err}}(g_{\mathrm{err}}(N)) = N$. We suppose that we can express the error in the primal and dual field solutions as

$$\|u^{N_t}(\boldsymbol{\mu}) - u^{N_t}_{N^{\mathrm{pr}}}(\boldsymbol{\mu})\|_X = \|u^{N_t}(\boldsymbol{\mu})\|_X g_{\mathrm{err}}(N^{\mathrm{pr}}) \tag{2.109}$$

and

$$\|\Psi^{N_t}(\boldsymbol{\mu}) - \Psi^{N_t}_{N^{\mathrm{du}}}(\boldsymbol{\mu})\|_X = \|\Psi^{N_t}(\boldsymbol{\mu})\|_X g_{\mathrm{err}}(N^{\mathrm{du}}). \tag{2.110}$$

From our conventions, it follows that for $N^{\mathrm{pr}} = 0$ and $N^{\mathrm{du}} = 0$, $g(0) = 1$. As we are using the same convergence function for both the primal and the dual approximation, we assume that their convergence behavior is similar. If this is not the case, we have to replace the general g_{err} by $g^{\mathrm{pr}}_{\mathrm{err}}$ and $g^{\mathrm{du}}_{\mathrm{err}}$. In addition, we shall suppose that - motivated by result (2.106) - the error in the RB output approximation can also be related to the norm of the FE primal and dual solution with the help of the convergence function:

$$|s^{N_t}(\boldsymbol{\mu}) - s^{N_t}_{N^{\mathrm{pr}},N^{\mathrm{du}}}(\boldsymbol{\mu})| = C_s\|u^{N_t}(\boldsymbol{\mu})\|_X\|\Psi^{N_t}(\boldsymbol{\mu})\|_X g_{\mathrm{err}}(N^{\mathrm{pr}})g_{\mathrm{err}}(N^{\mathrm{du}}), \tag{2.111}$$

where C_s is a constant that reflects the non-zero angle between the primal and dual errors.

If we want to reach a fixed output error $|s^{N_t}(\boldsymbol{\mu}) - s^{N_t}_{N^{\mathrm{pr}},N^{\mathrm{du}}}(\boldsymbol{\mu})| = C_s\|u^{N_t}(\boldsymbol{\mu})\|_X\|\Psi^{N_t}(\boldsymbol{\mu})\|_X\epsilon$, controlled by the parameter ϵ, then the associated online computational costs for the computation of $s^{N_t}_N(\boldsymbol{\mu})$ with the full primal-dual framework with the (in this case) optimal choice $N^{\mathrm{pr}} = N^{\mathrm{du}} = N$ are related to the online computational costs for the computation of $s^{N_t}_{N^*}(\boldsymbol{\mu})$, where only the primal approximation is taken into account ($N^{\mathrm{du}} = 0$, $N^{\mathrm{pr}} = N^* \neq 0$) as

$$\frac{\text{ONLINE COST OF } s^{N_t}_N(\boldsymbol{\mu}) \text{ WITH ADJOINT}}{\text{ONLINE COST OF } s^{N_t}_{N^*}(\boldsymbol{\mu}) \text{ WITHOUT ADJOINT}} = 2\left(\frac{g^{-1}_{\mathrm{err}}(\sqrt{\epsilon})}{g^{-1}_{\mathrm{err}}(\epsilon)}\right)^3; \tag{2.112}$$

In the "worst" case, when $g_{\mathrm{err}}(N) = e^{-\omega N}$ for a given positive real ω and thus the convergence is exponential, the ratio in (2.112) is $\frac{1}{4}$ (in fact, this case is optimal for the RB approximation but the application of the full primal-dual framework gives us the smallest savings compared to the pure primal approximation). We can thus obtain a reduction of 75% even in the "worst" case. For other, e.g. algebraic convergence rates, the computational savings are even higher and particularly pronounced for smaller ϵ [68].

If the convergence behavior of the primal and the dual approximation differ considerably, $N^{\mathrm{pr}} = N^{\mathrm{du}}$ is not the best choice anymore. The optimal choice of N^{pr}, N^{du} which minimizes the online effort will depend on the ratio between $g^{\mathrm{pr}}_{\mathrm{err}}$ and $g^{\mathrm{du}}_{\mathrm{err}}$. If the convergence of the primal approximation is slow, that means $g^{\mathrm{pr}}_{\mathrm{err}} \ll g^{\mathrm{du}}_{\mathrm{err}}$, we should choose $N^{\mathrm{pr}} \neq 0$, $N^{\mathrm{du}} = 0$ and vice versa. We should also note that if we do not want to compute only one output but many different outputs with different output functionals, the primal/dual approximation is no longer affordable as it requires the solution of a new dual problem for each of the outputs.

To conclude our discussion of the noncompliant problems, we consider the effect of the primal-dual framework on our a posteriori output error bounds. First, we consider the case where only the

primal problem is solved $N^{\mathrm{du}} = 0$. This corresponds to $\Psi_{N^{\mathrm{du}}}^{N_t} = 0$ in (2.108) and thus $s_{N^{\mathrm{pr}},N^{\mathrm{du}}}^{N_t}(\boldsymbol{\mu}) = l(u_{N^{\mathrm{pr}}}^{N_t}(\boldsymbol{\mu}))$ which gives us

$$|s^{N_t}(\boldsymbol{\mu}) - s_{N^{\mathrm{pr}},N^{\mathrm{du}}}^{N_t}(\boldsymbol{\mu})| = |l(u_{N^{\mathrm{pr}}}^{N_t}(\boldsymbol{\mu}) - u^{N_t}(\boldsymbol{\mu}))|. \tag{2.113}$$

We can conclude that the general form of our a posteriori error bound must then be

$$\Delta_N^s(\boldsymbol{\mu}) = \|l\|_{(X^{N_t})'}\Delta_N(\boldsymbol{\mu}), \tag{2.114}$$

where $\|l\|_{(X^{N_t})'}$ denotes the dual norm of l and $\Delta_N(\boldsymbol{\mu})$ is the error bound for $\|u^{N_t}(\boldsymbol{\mu}) - u_{N^{\mathrm{pr}}}^{N_t}(\boldsymbol{\mu})\|_X$.

Even though the output error bound (2.114) is indeed rigorous, the sharpness is very dissatisfactory in some situations. This shall be illuminated for the compliant case $l = f$. From (2.32) and (2.57) (extended to the X norm [55]) we get

$$\frac{\Delta_N^s(\boldsymbol{\mu})}{|s^{N_t}(\boldsymbol{\mu}) - s_{N^{\mathrm{pr}},N^{\mathrm{du}}}^{N_t}|} \geq \frac{\|l\|_{(X^{N_t})'}}{\gamma^e(\boldsymbol{\mu})\|u^{N_t}(\boldsymbol{\mu}) - u_{N^{\mathrm{pr}}}^{N_t}(\boldsymbol{\mu})\|_X}. \tag{2.115}$$

Unfortunately, if the RB output approaches the "truth" FE output $u_{N^{\mathrm{pr}}}^{N_t}(\boldsymbol{\mu}) \to u^{N_t}(\boldsymbol{\mu})$, the effectivity of the output error bound (2.114) tends to infinity. The problem is that the contribution of the test space to the convergence of the output is not reflected in the error bound (2.114). We anticipate that this problem exists for any l "close" to f [68].

These difficulties do not arise in the full primal-dual framework if the adjoint approximation is included in the computations and the error bound. This will ensure a stable limit in the case $N \to \infty$ (and consequently $u_{N^{\mathrm{pr}}}^{N_t}(\boldsymbol{\mu}) \to u^{N_t}(\boldsymbol{\mu})$). The error bound will then in the non-compliant case take the form

$$\Delta_N^s(\boldsymbol{\mu}) \equiv \frac{\|r^{\mathrm{pr}}(\cdot;\boldsymbol{\mu})\|_{(X^{N_t})'}}{(\alpha_{LB}(\boldsymbol{\mu}))^{1/2}} \frac{\|r^{\mathrm{du}}(\cdot;\boldsymbol{\mu})\|_{(X^{N_t})'}}{(\alpha_{LB}(\boldsymbol{\mu}))^{1/2}}, \tag{2.116}$$

where

$$r^{\mathrm{du}}(v;\boldsymbol{\mu}) = -l(v) + a(v, \Psi_{N^{\mathrm{du}}}^{N_t}(\boldsymbol{\mu});\boldsymbol{\mu}), \quad \forall v \in X^e \tag{2.117}$$

is the dual residual. That means that (2.116) is built as the product of the primal and the dual error. We now reflect the influence of both approximation spaces "through" $\Phi = \Psi_{N^{\mathrm{du}}}^{N_t}$ and hence can recover our earlier N-independent effectivity bound (2.57) in the compliant case. However, although these effectivities are bounded, they can get quite large in the non-compliant case when the correlation between primal and dual errors can be extremely weak.

2.6.2 Nonsymmetric Operators

The modifications in the formulation to consider for nonsymmetric operators are very small. As already mentioned, nonsymmetric problems are non-compliant and therefore should be treated by the full primal dual framework as $u^e(\boldsymbol{\mu}) \neq \Psi^e(\boldsymbol{\mu})$. The general approximation framework needs no additional changes, only the a posteriori error estimation is concerned. The output error bound is given by (2.116) which requires two dual norm calculations for the primal and the dual residual. The principal offline-online procedure remains unchanged. Only the SCM procedure to compute a lower bound for the coercivity constant $\alpha_{LB}(\boldsymbol{\mu})$ is changed a little. We simply apply the SCM procedure of section 2.5.2 to the symmetric part a_S of our non-symmetric bilinear form a:

$$a(u, v;\boldsymbol{\mu}) = a_S(u, v;\boldsymbol{\mu}) - a_{SS}(u, v;\boldsymbol{\mu}), \tag{2.118}$$

where

$$a_S(u, v;\boldsymbol{\mu}) = \frac{1}{2}(a(u, v;\boldsymbol{\mu}) + a(v, u;\boldsymbol{\mu})) \tag{2.119}$$

is the symmetric part of a and

$$a_{SS}(u, v; \boldsymbol{\mu}) = \frac{1}{2}(a(u, v; \boldsymbol{\mu}) - a(v, u; \boldsymbol{\mu})) \tag{2.120}$$

is the skew-symmetric part of a.

2.7 Affine Geometric Parametric Variations

This work concentrates on the application of the reduced basis methodology presented in the preceding sections in the shape optimization context. A crucial ingredient for this purpose is the family of geometric parametric variations that are consistent with the affine restriction (2.11). The motivation for the restriction on affine variations has already been given before: The affine parameter dependence allows us to construct very efficient offline-online decomposition procedures. In the following, the necessary fundamentals for affine geometric variations in the reduced basis context in the two-dimensional case will be presented. A more detailed introduction with many examples for different kinds of geometries can be found in [68].

2.7.1 Preconditions

At first glance, the reduced basis methodology described before and the shape optimization context (see section 6.5 and [39]) where we are going to work with parameter dependent geometries (that means the boundaries and thus the RB domain is parameter dependent) may not match: The RB recipe requires a parameter independent domain Ω as the snapshots we use for the construction of our basis functions have to be defined relative to the same spatial configuration. This difficulty can be resolved by interpreting Ω as parameter independent reference domain which is related to the parameter-dependent "actual" or "original" domain of interest $\Omega_o(\boldsymbol{\mu})$ via an affine mapping $T^{\text{aff}}(\boldsymbol{x}; \boldsymbol{\mu})$.

For this purpose, we first introduce a set of mutually non-overlapping open "regions" $R_{o,l}(\boldsymbol{\mu})$, $1 \leq l \leq L_{\text{reg}}$ such that

$$\Omega_o(\boldsymbol{\mu}) = \cup_{l=1}^{L_{\text{reg}}} R_{o,l}(\boldsymbol{\mu}), \tag{2.121}$$

and

$$R_{o,l}(\boldsymbol{\mu}) \cap R_{o,l'}(\boldsymbol{\mu}) = \emptyset, \quad 1 \leq l < l' \leq L_{\text{reg}}. \tag{2.122}$$

These regions can e.g. represent different material properties, but they can also be used for algorithmic purposes to ensure well-behaved mappings. We will come back to this issue in chapter 3.1. The boundaries of $R_{o,l}(\boldsymbol{\mu})$, which do not reside on the boundary of $\Omega_o(\boldsymbol{\mu})$, will be called *internal interfaces*.

We can then introduce a domain decomposition of $\Omega_o(\boldsymbol{\mu})$,

$$\Omega_o(\boldsymbol{\mu}) = \cup_{k=1}^{K_{\text{dom}}} \Omega_o^k(\boldsymbol{\mu}), \tag{2.123}$$

which consists of mutually non-overlapping open subdomains $\Omega_o^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{\text{dom}}$,

$$\Omega_o^k(\boldsymbol{\mu}) \cap \Omega_o^{k'}(\boldsymbol{\mu}) = \emptyset, \quad 1 \leq k < k' \leq K_{\text{dom}}. \tag{2.124}$$

This domain decomposition has to respect the previously introduced regions in the sense that

$$R_{o,l}(\boldsymbol{\mu}) = \cup_{k \in \mathbb{K}_l} \Omega_k^l, \tag{2.125}$$

where the $\mathbb{K}_l$, $1 \leq l \leq L_{\text{reg}}$ are mutually exclusive subsets of $\{1, \ldots, K_{\text{dom}}\}$.

Our reference domain is then simply defined for a reference parameter value $\boldsymbol{\mu}_{\text{ref}} \in D$ as $\Omega \equiv$

$\Omega_o(\boldsymbol{\mu}_{\text{ref}})$. In the following we will identify $\Omega^k = \Omega_o^k(\boldsymbol{\mu}_{\text{ref}})$, $1 \le k \le K_{\text{dom}}$ and $R_l = R_{o,l}(\boldsymbol{\mu}_{\text{ref}})$, $1 \le l \le L_{\text{reg}}$ for brevity.

The "K_{dom}" domain decomposition of Ω shall be denoted our "RB triangulation". It will play an important role in the generation of our affine representation (2.11). The very fine N_t FE mesh has to be a subtriangulation of the RB triangulation. Both the FE and RB approximations are defined over the reference domain. The choice of $\boldsymbol{\mu}_{\text{ref}}$ has an influence on the accuracy of the underlying FE approximation as it controls the distortion of the mesh for the actual domains.

As mentioned before, the original domain (resp. the original subdomains) and the reference domain (resp. the reference subdomains) must be connectable via an affine mapping $T^{\text{aff}}(\cdot;\boldsymbol{\mu}) : \Omega^k \to \Omega_o^k(\boldsymbol{\mu})$, $1 \le k \le K_{\text{dom}}$:

$$\Omega_o^k(\boldsymbol{\mu}) = T^{\text{aff},k}(\Omega^k;\boldsymbol{\mu}), \quad 1 \le k \le K_{\text{dom}}. \tag{2.126}$$

The affine mappings must be individually bijective and collectively continuous, that means they have to fulfill the following interface condition:

$$T^{\text{aff},k}(\boldsymbol{x};\boldsymbol{\mu}) = T^{\text{aff},k'}(\boldsymbol{x};\boldsymbol{\mu}), \quad \forall \boldsymbol{x} \in \Omega^k \cap \Omega^{k'}, \quad 1 \le k < k' \le K_{\text{dom}}; \tag{2.127}$$

For reasons of computational efficiency it is important to note that K_{dom} is defined with respect to the exact problem and therefore does not depend on N_t.

The concrete affine transformations are then given for $1 \le k \le K_{\text{dom}}$, for any $\boldsymbol{\mu} \in D$ and for any $\boldsymbol{x} \in \Omega^k$ as

$$T_i^{\text{aff},k}(\boldsymbol{x},\boldsymbol{\mu}) = C_i^{\text{aff},i}(\boldsymbol{\mu}) + \sum_{j=1}^{d} G_{ij}^{\text{aff},k}(\boldsymbol{\mu}) x_j, \quad 1 \le i \le d, \tag{2.128}$$

for given translation vectors $C^{\text{aff},k} : D \to \mathbb{R}^d$ and linear transformation matrices $G^{\text{aff},i} : D \to \mathbb{R}^{d\times d}$. The linear transformation matrices can effect rotation, scaling and/or shear and have to be invertible. The associated Jacobians can be defined as

$$J^{\text{aff},k}(\boldsymbol{\mu}) = |\det\left(G^{\text{aff},k}(\boldsymbol{\mu})\right)|, \quad 1 \le k \le K_{\text{dom}}. \tag{2.129}$$

For invertible mappings they are strictly positive.

We note that the interface condition (2.127) allows us to interpret the set of local mappings as a global bijective piecewise affine transformation $T^{\text{aff}}(\cdot;\boldsymbol{\mu}) : \Omega \to \Omega_o(\boldsymbol{\mu})$. This global mapping is then given for any $\boldsymbol{\mu} \in D$ by

$$T^{\text{aff}}(\boldsymbol{x};\boldsymbol{\mu}) = T^{\text{aff},k}(\boldsymbol{x};\boldsymbol{\mu}), \quad k = \min_{k' \in \{1,\ldots,K_{\text{dom}}\} | \boldsymbol{x} \in \Omega^{k'}} k'. \tag{2.130}$$

2.7.2 Affine Mappings for a Single Subdomain

In this section we will go more into the details of the technology to define our affine mappings and present the basic building blocks of our RB triangulation that allow well-defined affine transformations. As for these purposes it is sufficient to concentrate on a single subdomain, we shall suppress the subdomain superscript for clarity of exposition. The matrices $C^{\text{aff}}(\boldsymbol{\mu}) \in \mathbb{R}^d$ and $G^{\text{aff}}(\boldsymbol{\mu}) \in \mathbb{R}^{d\times d}$ in (2.128) are now called "mapping coefficients".

We will now recall some of the properties of affine transformations in two dimensions. First, straight lines are mapped to straight lines, parallelism is preserved and parallel lines of equal length are also mapped on parallel lines of equal length. Consequently, a parallelogram is mapped to a parallelogram and hence a triangle maps to a triangle. Second, an affine transformation maps ellipses to ellipses. These features will be exploited in the following for the development of a domain decomposition technique that is suitable for the RB context.

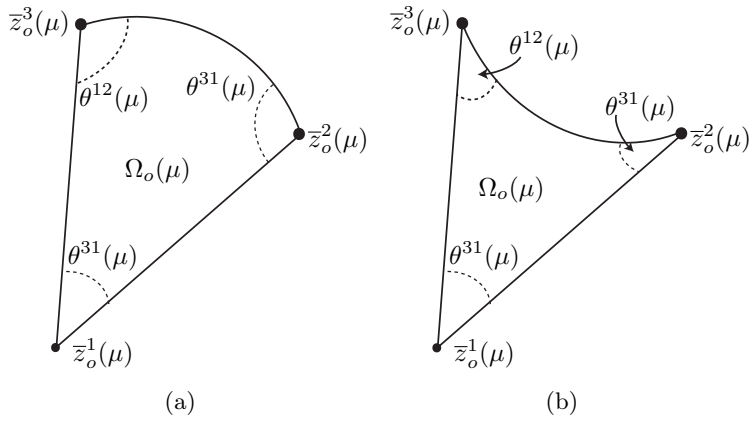


Figure 2.2: (a) "Inwards" elliptical triangle and (b) "outwards" elliptical triangle.

Our affine mapping contains in the two-dimensional case ($d = 2$) ($d(d+1)) = 6$ degrees of freedom, the mapping coefficients. It is therefore sufficient, for any given $\boldsymbol{\mu} \in D$, to consider the relationship between three non-collinear pre-image points in Ω, $(\boldsymbol{z}^1, \boldsymbol{z}^2, \boldsymbol{z}^3)$ and three parametrized image nodes in $\Omega_o(\boldsymbol{\mu})$, $(\boldsymbol{z}_o^1(\boldsymbol{\mu}), \boldsymbol{z}_o^2(\boldsymbol{\mu}), \boldsymbol{z}_o^3(\boldsymbol{\mu}))$. Note that every point consists of two components (z_1^i, z_2^i), $1 \le i \le 3$ resp. $(z_{o\,1}^i, z_{o\,2}^i)$, $1 \le i \le 3$ and therefore the application of (2.128) to these points constitutes a system of six independent equations to determine the six mapping coefficients:

$$z_{o\,i}^m(\boldsymbol{\mu}) = C_i^{\text{aff}}(\boldsymbol{\mu}) + \sum_{j=1}^{2} G_{ij}^{\text{aff}}(\boldsymbol{\mu}) z_j^m, \quad 1 \le i \le 2, \quad 1 \le m \le 3; \tag{2.131}$$

The assumption that the affine transformation is bijective thereby ensures that the image nodes are perforce also non-collinear (if the pre-image nodes are non-collinear) and hence the equations are perforce linear independent.

Our RB triangulation shall be built of (standard) triangles, elliptical triangles and general "curvy" triangles. They admit symbolic and numerical automation and are therefore the building blocks of choice in the rbMIT software package [23] that we will use for the reduced basis computations in this work. This software, the techniques applied and its usage will be presented in chapter 3. The three basic building block types are now discussed in detail.

Standard Triangles

The proceeding in the case of a standard triangle subdomain is clear - the three vertices of the triangle in the reference domain shall serve as pre-image nodes while the three vertices of the triangle in the actual ($\boldsymbol{\mu}$-dependent) domain shall serve as image nodes. In this case, our three points uniquely define not only the transformation but also the reference domain and parametrized domains. We recall that the pre-image nodes are obtained as the image nodes for a particular value of the parameter $\boldsymbol{\mu}_{\text{ref}}$. We can then readily establish the system of six linear equations to determine the six unknown mapping coefficients. In this way, we can construct an affine transformation from any reference triangle in $\mathbb{R}^2$ onto any desired triangle in $\mathbb{R}^2$. We note that it is not obligatory to choose the vertices of the triangles as our nodes defining the transformation, other characteristic points e.g. the barycentric coordinates of the FE context are also possible.

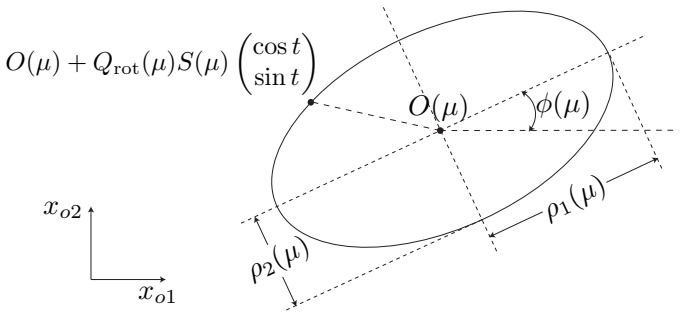


Figure 2.3: Definition of a point on a prescribed parametrized ellipse.

Elliptical Triangles

The second class of building blocks is the class of elliptical triangles. This class already covers a much greater range of possible geometries and the theory needed to describe well-defined and well-behaving elliptical triangles is in great parts also necessary for the more general curvy triangles. We can distinguish two different kinds of elliptic triangles: "inwards" and "outwards" triangles. Both types are depicted in Figure 2.2. In both cases, the elliptical triangle $\Omega_o(\boldsymbol{\mu})$ is defined by the three vertices $\boldsymbol{z}_o^1(\boldsymbol{\mu})$, $\boldsymbol{z}_o^2(\boldsymbol{\mu})$, $\boldsymbol{z}_o^3(\boldsymbol{\mu})$, the two straight lines $\overline{\boldsymbol{z}_o^1(\boldsymbol{\mu})\boldsymbol{z}_o^2(\boldsymbol{\mu})}$ and $\overline{\boldsymbol{z}_o^1(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}$ as well as the elliptical arc $\overline{\boldsymbol{z}_o^2(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}^{\text{arc}}$. We shall now precise the definition and description of the elliptical arc and explain the constraints that must be met by the location of the third point $\boldsymbol{z}_o^1(\boldsymbol{\mu})$ to ensure "proper" triangles and a continuous and well-defined global mapping in the multidomain context.

First, the description of the elliptical arc shall be derived from the definition of a parametrized ellipse as depicted in Figure 2.3. The ellipse is described implicitly by

$$(\boldsymbol{x}_o - O(\boldsymbol{\mu}))^T Q_{\text{rot}}(\boldsymbol{\mu}) S^{-2}(\boldsymbol{\mu}) Q_{\text{rot}}(\boldsymbol{\mu})^T (\boldsymbol{x}_o - O(\boldsymbol{\mu})) = 1. \tag{2.132}$$

A particular point on this ellipse is then given by

$$\boldsymbol{x}_o \equiv \begin{pmatrix} x_{o1} \\ x_{o2} \end{pmatrix} = O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \tag{2.133}$$

for given $t \in \mathbb{R}$. As we can see in Figure 2.3, $O(\boldsymbol{\mu}) : D \to \mathbb{R}^2$ is the center of the ellipse, $\rho_1(\boldsymbol{\mu}) : D \to \mathbb{R}_+$ and $\rho_2 : D \to \mathbb{R}_+$ define the lengths of the semi-axes of the ellipse and $\phi(\boldsymbol{\mu}) : D \to \mathbb{R}$ is the angle of inclination. With these quantities, the scaling matrix

$$S(\boldsymbol{\mu}) \equiv \begin{pmatrix} \rho_1(\boldsymbol{\mu}) & 0 \\ 0 & \rho_2(\boldsymbol{\mu}) \end{pmatrix} \tag{2.134}$$

and the rotation matrix

$$Q_{\text{rot}}(\boldsymbol{\mu}) = \begin{pmatrix} \cos\phi(\boldsymbol{\mu}) & -\sin\phi(\boldsymbol{\mu}) \\ \sin\phi(\boldsymbol{\mu}) & \cos\phi(\boldsymbol{\mu}) \end{pmatrix} \tag{2.135}$$

can be defined.

The description of the elliptical arc with these means is then as follows:

$$\overline{\boldsymbol{z}_o^2(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}^{\text{arc}} = \left\{ O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \,\middle|\, t_2 \le t \le t_3 \right\}. \tag{2.136}$$

with $t_2 \in \mathbb{R}$ and $t_3 \in \mathbb{R}$ chosen such that the points $\boldsymbol{z}_o^2(\boldsymbol{\mu})$ and $\boldsymbol{z}_o^3(\boldsymbol{\mu})$ are given as the endpoints of

the elliptical arc for $t = t_2$ resp. $t = t_3$:

$$\boldsymbol{z}_o^m(\boldsymbol{\mu}) = O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu}) \begin{pmatrix} \cos t_m \\ \sin t_m \end{pmatrix}, \quad m = 2,\, 3. \tag{2.137}$$

In addition, we have to make sure that $0 \leq t_3 - t_2 < \pi$.

It remains to specify the location of the third point $\boldsymbol{z}_o^1(\boldsymbol{\mu})$. For elliptical triangles, this location is not arbitrary but has to be chosen in a way that ensures that the affine transformation generates the desired elliptical arc (2.136). This guarantees a continuous global mapping. In addition, to obtain well-defined elliptical triangles and consequently a well defined domain in the multidomain context, several internal angle conditions have to be met by our choice for $\boldsymbol{z}_o^1(\boldsymbol{\mu})$: $0 < \Theta^* < \pi,\ \forall \Theta^* \in \{\Theta^{12}, \Theta^{23}, \Theta^{31}\}$.

The first requirement can be fulfilled by the choice

$$\boldsymbol{z}_o^1(\boldsymbol{\mu}) = O(\boldsymbol{\mu}) + \omega Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu}) \begin{pmatrix} \cos t_1 \\ \sin t_1 \end{pmatrix}, \tag{2.138}$$

for given $\omega \in \mathbb{R}$ and $t_1 \in [t_2, t_3]$. As the choice $\omega = 1$ gives us an equation equivalent to (2.137), the three corner points can be expressed as

$$\boldsymbol{z}_o^m(\boldsymbol{\mu}) = O(\boldsymbol{\mu}) + \omega_m Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu}) \begin{pmatrix} \cos t_m \\ \sin t_m \end{pmatrix}, \quad 1 \leq m \leq 3, \tag{2.139}$$

for $\omega_2 = \omega_3 = 1$. Consequently, the pre-image points are given as

$$\boldsymbol{z}_o^m(\boldsymbol{\mu}_{\text{ref}}) = O(\boldsymbol{\mu}_{\text{ref}}) + \omega_m Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})S(\boldsymbol{\mu}_{\text{ref}}) \begin{pmatrix} \cos t_m \\ \sin t_m \end{pmatrix}, \quad 1 \leq m \leq 3. \tag{2.140}$$

From these representations we can identify our affine mapping as

$$\begin{aligned} \boldsymbol{z}_o^m(\boldsymbol{\mu}) &= C^{\text{aff}}(\boldsymbol{\mu}) + G^{\text{aff}}(\boldsymbol{\mu})\boldsymbol{z}^m \\ &= (O(\boldsymbol{\mu}) - Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu})S(\boldsymbol{\mu}_{\text{ref}})^{-1}Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})^T O(\boldsymbol{\mu}_{\text{ref}})) \\ &\quad + (Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu})S(\boldsymbol{\mu}_{\text{ref}})^{-1}Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})^T)\boldsymbol{z}^m. \end{aligned} \tag{2.141}$$

As for both the inwards and outwards cases Ω is perforce star-shaped with respect to $\boldsymbol{z}^1$, we can write

$$\Omega = \bigcup_{t \in [t_2, t_3]} \overline{\boldsymbol{z}^1 \left(O(\boldsymbol{\mu}_{\text{ref}}) + Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})S(\boldsymbol{\mu}_{\text{ref}}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \right)}. \tag{2.142}$$

With (2.141) we then obtain the definition of our elliptical triangles:

$$\begin{aligned} \Omega_o(\boldsymbol{\mu}) &= T^{\text{aff}} \bigcup_{t \in [t_2, t_3]} \overline{\boldsymbol{z}^1 \left(O(\boldsymbol{\mu}_{\text{ref}}) - Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})S(\boldsymbol{\mu}_{\text{ref}}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \right)} \\ &= \bigcup_{t \in [t_2, t_3]} \overline{T^{\text{aff}} \boldsymbol{z}^1 \left(O(\boldsymbol{\mu}_{\text{ref}}) - Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})S(\boldsymbol{\mu}_{\text{ref}}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \right)} \\ &= \bigcup_{t \in [t_2, t_3]} \overline{\boldsymbol{z}_o^1 \left(O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \right)} \end{aligned} \tag{2.143}$$

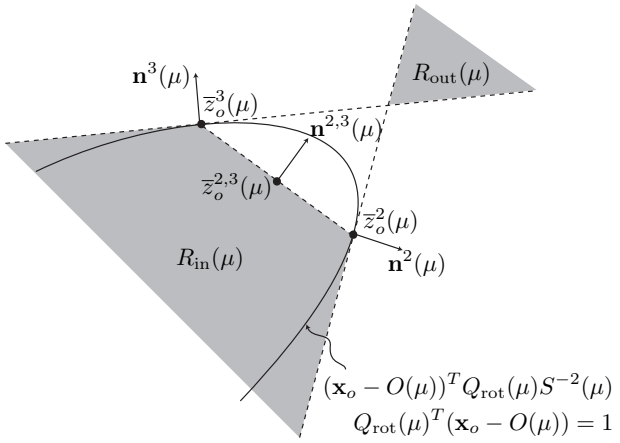


Figure 2.4: Regions in which $z_o^1(\boldsymbol{\mu})$ must reside for an elliptical triangle in the inwards case ($R_{\text{in}}(\boldsymbol{\mu})$) and the outwards case ($R_{\text{out}}(\boldsymbol{\mu})$).

as (remember that $Q_{\text{rot}}(\boldsymbol{\mu})$ is orthogonal)

$$\begin{aligned}
&T^{\text{aff}}\left(O(\boldsymbol{\mu}_{\text{ref}}) - Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})S(\boldsymbol{\mu}_{\text{ref}})\begin{pmatrix}\cos t\\ \sin t\end{pmatrix}\right)\\
&\quad = O(\boldsymbol{\mu}) - Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu})S(\boldsymbol{\mu}_{\text{ref}})^{-1}Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})^T O(\boldsymbol{\mu}_{\text{ref}})\\
&\quad + Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu})S(\boldsymbol{\mu}_{\text{ref}})^{-1}Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})^T\left(O(\boldsymbol{\mu}_{\text{ref}}) + Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})S(\boldsymbol{\mu}_{\text{ref}})\begin{pmatrix}\cos t\\ \sin t\end{pmatrix}\right)\\
&= O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu})S(\boldsymbol{\mu})\begin{pmatrix}\cos t\\ \sin t\end{pmatrix}.
\end{aligned} \tag{2.144}$$

The second requirement - the internal angle conditions - is illustrated in Figure 2.4 [68]. In the inwards case, a necessary and sufficient condition to ensure the internal angle conditions

$$0 < \Theta^* < \pi \quad \forall \Theta^* \in \{\Theta^{12}, \Theta^{23}, \Theta^{31}\} \tag{2.145}$$

is given for an inwards elliptical triangle by

$$\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in R_{\text{in}}(\boldsymbol{\mu}) \tag{2.146}$$

where

$$\begin{aligned}
R_{\text{in}}(\boldsymbol{\mu}) = \{\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in \mathbb{R}^2 | &(\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^2(\boldsymbol{\mu}))^T \boldsymbol{n}^2(\boldsymbol{\mu}) && < 0,\\
&(\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^3(\boldsymbol{\mu})))^T \boldsymbol{n}^3(\boldsymbol{\mu}) && < 0,\\
&(\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^{2,3}(\boldsymbol{\mu}))^T \boldsymbol{n}^{2,3}(\boldsymbol{\mu}) && < 0\},
\end{aligned} \tag{2.147}$$

and for the outwards elliptical triangle by

$$\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in R_{\text{out}}(\boldsymbol{\mu}), \tag{2.148}$$

where

$$\begin{aligned}
R_{\text{out}}(\boldsymbol{\mu}) = \{\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in \mathbb{R}^2 | &(\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^2(\boldsymbol{\mu}))^T \boldsymbol{n}^2(\boldsymbol{\mu}) && > 0,\\
&(\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^3(\boldsymbol{\mu}))^T \boldsymbol{n}^3(\boldsymbol{\mu}) && > 0\}.
\end{aligned} \tag{2.149}$$

Here $\boldsymbol{n}^2(\boldsymbol{\mu})$ and $\boldsymbol{n}^3(\boldsymbol{\mu})$ are the outwards-facing normals to the ellipse at $\boldsymbol{z}^2(\boldsymbol{\mu})$ and $\boldsymbol{z}^3(\boldsymbol{\mu})$ respectively, $\boldsymbol{z}_o^{2,3}(\boldsymbol{\mu}) = \frac{1}{2}(\boldsymbol{z}_o^2(\boldsymbol{\mu}) + \boldsymbol{z}_o^3(\boldsymbol{\mu}))$ and $\boldsymbol{n}^{2,3}(\boldsymbol{\mu})$ is the "outwards-facing" normal to the line segment $\overline{\boldsymbol{z}_o^2(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}$ at $\boldsymbol{z}_o^{2,3}(\boldsymbol{\mu})$.

We note here that for elliptical triangles it is possible to derive explicit conditions on ω such that the internal angle conditions (2.147) and(2.149) are satisfied, for details see [68]. These conditions are independent of $\boldsymbol{\mu}$. An important feature of the elliptical triangles is that they are consistent under refinement, that means that if we split an elliptical triangle for which the internal angle conditions (2.147) and (2.149) are fulfilled, the resulting two daughter elliptical triangles also satisfy the internal angle conditions.

Curvy Triangles

To enlarge the possible range of geometries even more, the elliptical triangles are extended to "curvy" triangles. This is done by replacing $(\cos t, \sin t)^T$ in (2.139) with a general parametrization $(g_1(t), g_2(t))^T$. The curvy arcs have to be either strictly convex ("inwards") or strictly concave ("outwards") for all $\boldsymbol{\mu} \in D$. It is easy to evaluate this property by the sign of the derivative of the normal or the curvature. We can then directly apply the internal angle conditions (2.147) and (2.149) that we have stated for the elliptical case. Under certain circumstances, they can be reduced to a simple set of algebraic equations in terms of the functions $g_1(t)$ and $g_2(t)$ and their derivatives evaluated at t_1, t_2 and t_3. The numerical solution of this set of equations is feasible. However, it is not possible to derive a simple closed form as it is the case for the elliptical triangles.

2.7.3 Piecewise-Affine Mappings for Multiple Subdomains

To treat more complex geometries, it is necessary to allow our domain to be built of several (standard, elliptical or curvy) triangles. We are then not restricted to a single affine mapping, but result in a piecewise affine mapping based on this domain decomposition. We can thus consider geometrical domains and regions for which the boundary and internal interfaces can be represented either by straight edges or by elliptical respectively curvy triangles.

The multi-domain mapping process is then performed in three steps. First, the RB triangulation is generated on the reference domain Ω together with the associated reference regions. The RB triangulation has to be compatible with the mapping continuity condition (2.127) and all elliptical and curvy subtriangles have to be well-defined and fulfill the internal angle conditions (2.147) and (2.149). The procedure applied to generate this RB triangulation in the rbMIT software package [23] will be discussed in detail in section 3.1. In the second step, the necessary parameter-dependent affine mappings for each subdomain are constructed, as described in the previous section. In the last step we have to translate the parametric mappings obtained for each subdomain into PDE coefficients. These terms further have to be optimized, that means similar PDE coefficients should be summarized into single terms, to arrive at an economical affine expansion (2.11). We will describe this part of the procedure in the next section.

2.7.4 Bilinear Forms for Affine Geometric Parametric Variations

This section is dedicated to the last question that is still open: How can we use the affine mappings derived in the previous sections to get an affine representation (2.11) of our problem (which has a parameter-dependent geometry) on a parameter-independent reference geometry. We will first address the transformation of the formulation on the original domain to the formulation on the reference domain and then explain how the affine representation can be derived from this formulation.

From the Original Domain to the Reference Domain

If we consider a problem analogue to (2.1) with a parameter-dependent domain $\Omega_o(\boldsymbol{\mu})$ which realizes the affine geometry precondition as described in the previous section, this problem can be written in general form as: Given $\boldsymbol{\mu} \in D$, we evaluate

$$s_o^e(\boldsymbol{\mu}) = l_o(u_o^e(\boldsymbol{\mu})), \tag{2.150}$$

where $u_o^e(\boldsymbol{\mu}) \in X_o^e(\boldsymbol{\mu})$ satisfies

$$a_o(u_o^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f_o(v), \quad \forall v \in X_o^e(\boldsymbol{\mu}). \tag{2.151}$$

For simplicity we assume that we have homogeneous Dirichlet boundary conditions over the entire boundary, which corresponds to $X_o^e(\boldsymbol{\mu}) = H_0^1(\Omega_o(\boldsymbol{\mu}))$.

A sufficient condition on a_o that ensures an affine expansion of the bilinear form (if the affine geometry precondition is fulfilled) is fulfilled if $a_o(\cdot, \cdot; \boldsymbol{\mu}) : H^1(\Omega_o(\boldsymbol{\mu})) \times H^1(\Omega_o(\boldsymbol{\mu})) \to \mathbb{R}$ can be expressed as

$$a_o(w, v; \boldsymbol{\mu}) = \sum_{l=1}^{L_{\text{reg}}} \int_{\cup_{k \in \mathbb{K}_l} \Omega_o^k(\boldsymbol{\mu})} \begin{bmatrix} \frac{\partial w}{\partial x_{o1}} & \frac{\partial w}{\partial x_{o2}} & w \end{bmatrix} K_{o,l}(\boldsymbol{\mu}) \begin{bmatrix} \frac{\partial v}{\partial x_{o1}} \\ \frac{\partial v}{\partial x_{o2}} \\ v \end{bmatrix}. \tag{2.152}$$

The matrices $K_{o,l} : D \to \mathbb{R}^{3\times 3}$, $1 \le l \le L_{\text{reg}}$ are in the symmetric case symmetric positive definite matrices where the components $K_{o,l\,1,3}$, $K_{o,l\,2,3}$, $K_{o,l\,3,1}$ and $K_{o,l\,3,2}$ are equal zero. In the non-symmetric case, all components may be non-zero. A similar requirement may be posed on $f_o : H^1(\Omega_o(\boldsymbol{\mu})) \to \mathbb{R}$. We require that it can be expressed as

$$f_o(v) = \sum_{l=1}^{L_{\text{reg}}} \int_{\cup_{k \in \mathbb{K}_l} \Omega_o^k(\boldsymbol{\mu})} F_{o,l}(\boldsymbol{\mu}) v, \tag{2.153}$$

with $F_{o,l} : D \to \mathbb{R}$, $\quad 1 \le l \le L_{\text{reg}}$. We note that these conditions are sufficient but not necessary and can be relaxed in some cases [68].

To transform this formulation on the reference domain to recover (2.1), we first identify $s^e(\boldsymbol{\mu}) = s_o^e(\boldsymbol{\mu})$ and $u^e(\boldsymbol{\mu}) = u_o^e(\boldsymbol{\mu}) \circ T^{\text{aff}}(\cdot; \boldsymbol{\mu})$. We then recall that

$$\frac{\partial}{\partial x_{oi}} = \frac{\partial x_j}{\partial x_{oi}} \frac{\partial}{\partial x_j} = (G^{\text{aff},k}(\boldsymbol{\mu}))^{-1}_{1i} \frac{\partial}{\partial x_1} + (G^{\text{aff},k}(\boldsymbol{\mu}))^{-1}_{2i} \frac{\partial}{\partial x_2} \quad i = 1, 2, \tag{2.154}$$

in $\Omega_o^k(\boldsymbol{\mu})$ and

$$d\Omega_o^k(\boldsymbol{\mu}) = J^{\text{aff},k}(\boldsymbol{\mu}) d\Omega. \tag{2.155}$$

It then follows that the transformed bilinear form a can be expressed as

$$a(w, v; \boldsymbol{\mu}) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega^k} \begin{bmatrix} \frac{\partial w}{\partial x_1} & \frac{\partial w}{\partial x_2} & w \end{bmatrix} K^k(\boldsymbol{\mu}) \begin{bmatrix} \frac{\partial v}{\partial x_1} \\ \frac{\partial v}{\partial x_2} \\ v \end{bmatrix}. \tag{2.156}$$

The $K^k : D \to \mathbb{R}^{3\times 3}$, $1 \le k \le K_{\text{dom}}$ are given by

$$K^k(\boldsymbol{\mu}) = J^{\text{aff},k}(\boldsymbol{\mu}) G^k(\boldsymbol{\mu}) K_{o,l}(\boldsymbol{\mu}) (G^k(\boldsymbol{\mu}))^T, \quad k \in \mathbb{K}_l, \tag{2.157}$$

for $1 \le l \le L_{\text{reg}}$. Recall that for $1 \le l \le L_{\text{reg}}$, $\mathbb{K}_l$ is the set of subdomains associated with region l

(2.125). The $G^k : D \to \mathbb{R}^{3\times 3}$ are given by

$$G^k(\boldsymbol{\mu}) = \begin{pmatrix} (G^{\mathrm{aff},k}(\boldsymbol{\mu}))^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}, \quad 1 \le k \le K_{\mathrm{dom}}. \tag{2.158}$$

The transformed linear form can be expressed similarly as

$$f(v) = \sum_{k=1}^{K_{\mathrm{dom}}} \int_{\Omega_k} F^k(\boldsymbol{\mu}) v, \tag{2.159}$$

where $F^k : D \to \mathbb{R}$, $1 \le k \le K_{\mathrm{dom}}$ is given by

$$F^k = J^{\mathrm{aff},k}(\boldsymbol{\mu}) F_{o,l}(\boldsymbol{\mu}), \quad \forall k \in \mathbb{K}_l, \quad 1 \le l \le L_{\mathrm{reg}}. \tag{2.160}$$

In general, the $K^k(\boldsymbol{\mu})$ and $F^k(\boldsymbol{\mu})$ will be different for each subdomain Ω^k.

Affine Formulation

The affine formulation of (2.156) can then easily be derived by simply expanding this expression (in terms of the subdomains Ω^k and the different entries of K_{ij}^k, $1 \le i, j \le 3$, $1 \le k \le K_{\mathrm{dom}}$. This results in

$$a(w, v; \boldsymbol{\mu}) = K_{11}^1(\boldsymbol{\mu}) \int_\Omega^1 \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_1} + K_{12}^1(\boldsymbol{\mu}) \int_\Omega^1 \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_2} + \cdots + K_{33}^{K_{\mathrm{dom}}}(\boldsymbol{\mu}) \int_\Omega^{K_{\mathrm{dom}}} wv. \tag{2.161}$$

The affine representation is now clear: For each term in (2.161) the (parameter-independent) integral represents $a^q(w, v)$, while the (parameter-dependent) prefactor represents $\Theta^q(\boldsymbol{\mu})$. The linear form f admits a similar treatment.

The affine representation obtained by this process contains at most $Q_a = 9K_{\mathrm{dom}}$ terms. We can considerably decrease this number by using optimization strategies. First, we note that in the case of symmetric bilinear forms, the number of affine terms is decreased to at most $Q_a = 4K_{\mathrm{dom}}$ terms due to the special structure of $K_{o,l}(\boldsymbol{\mu})$ (and consequently $K^k(\boldsymbol{\mu})$) in the symmetric case. In some special cases the number of nonzero terms in (2.161) is even reduced to $Q_a = 2K_{\mathrm{dom}}$. In other situations, many terms can be economized if linear dependent entries are "put together". This can be illustrated by the following example. If we assume that in our development (2.161) $K_{11}^2(\boldsymbol{\mu}) = \mathrm{Const}\ K_{11}^1(\boldsymbol{\mu})$, we may reduce Q_a by one term by redefining

$$a^1(w, v) = \int_{\Omega_1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_1} + \mathrm{Const} \int_{\Omega_2} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_1}. \tag{2.162}$$

Another possibility to reduce the number of terms Q_a is an intelligent choice of user-provided initial control points and edges for the RB triangulation. This can help to exploit symmetry effects and isolate geometric variation. We will come back to this issue in the practical part of this work.

The last point to remember is the admissible classes of boundary conditions. Dirichlet conditions as well as homogeneous Neumann conditions can be prescribed on any straight, circular, elliptical or general curvy edge of the domain boundary. Inhomogeneous Neumann conditions or Robin conditions however may be prescribed only on straight or circular edges, as they modify f and a, respectively. If they are prescribed on elliptical or general curvy edges, the mapping techniques introduced in the preceding sections do not admit an affine form (2.11) and it would be necessary to invoke non-affine techniques for the reduced-basis methodology [7, 67].

3 Software, Geometries and Flow Models

In this thesis, we have applied the reduced basis method to fluid flow around airfoil profiles. For all computations, the rbMIT software package [23] has been used. This software package has been developed in A.T. Patera's group at MIT during the last couple of years. The abilities, usage and mode of operation of this software will be presented in section 3.1. To investigate the performance of the software for problems with different levels of complexity, we worked with parametrized airfoil profiles of the NACA 4-digits family in several different geometric setups. The description of the airfoils and setups will be given in section 3.2. This chapter concludes in section 3.3 with a discussion of the different simplified fluid flow models that have been applied in this work which are used to further decrease the size/complexity of our problem by reducing the complexity of the physical properties of the flow.

3.1 The rbMIT Software Package

The rbMIT software package is designed for the solution of parametrized PDEs with the reduced basis method and is fully implemented in MATLAB. It consists of two parts. The first part is the kernel of the software which contains all methodology and algorithms necessary for the numerical solution for a given Galerkin FE system of partial differential equations by the RB method. That means that in this case the user has to provide the affine decomposition (2.11) of the underlying "truth" FE stiffness matrices and vectors in (2.2). This "low level" part of the software is therefore very flexible but requires much effort on user side, as the task of setting up the underlying FE problem and finding an affine decomposition is not trivial and very error-prone if done by hand (especially in the shape optimization context with parameter-dependent geometries). Some parts of this preliminary work can be done by the help of standard FE software, for example the MATLAB PDE Toolbox® or COMSOL Multiphysics™. More information about the usage of this RB kernel software can be found in [55].

The other part of the software is a symbolic preprocessor based on the RB kernel. It requires relatively few user input and is able to do most of the work automatically. The input required is on one hand the parametrized geometry of the problem and on the other hand the equation to be solved together with boundary conditions and the desired outputs. In addition, some (very few) control parameters have to be set to define the size and type of the underlying FE mesh and the possible parameter range and reference parameters. Other parameters to control the behavior of the algorithms in the offline reduced basis generation process can be set, but this requires more insight into the structure of the software. The mode of operation of the preprocessor is not numerical but symbolical - therefore we have no loss of accuracy in the preprocessing step, but for complex problems it can get computationally very expensive. The output of the symbolic preprocessor - the affine decomposition of the underlying FE system matrices and vectors together with the FE mesh - are then processed further in the low level RB kernel. This makes the rbMIT software package a very powerful tool that covers a wide range of problems, both concerning the relatively big class of PDE problems that can be handled and the classes of geometries that can be treated. A manual for the rbMIT software based on the high level symbolic preprocessor is available in [22]. In the next sections, the different user inputs and possible geometries and PDE problems will be presented, followed by a short explanation of the subsequent steps of the program in the offline and online stage.

3.1.1 Geometry Description and Processing

The software is based on an affine decomposition of the whole geometry into the three basic building blocks presented in section 2.7.2: standard triangles, elliptical triangles and more general curvy triangles. The geometry has to be provided by the user by a set of points (can be parameter dependent) and edges between those points, specifying different regions which are put together to form the entire region. Not only the domain boundaries but also internal interfaces and points can be defined. The edges can be either straight edges or parameter dependent curved lines, which have to be expressed in the following way:

$$\begin{pmatrix} x_{o1} \\ x_{o2} \end{pmatrix} = \begin{pmatrix} O_1(\boldsymbol{\mu}) \\ O_2(\boldsymbol{\mu}) \end{pmatrix} + \begin{pmatrix} \cos\phi(\boldsymbol{\mu}) & -\sin\phi(\boldsymbol{\mu}) \\ \sin\phi(\boldsymbol{\mu}) & \cos\phi(\boldsymbol{\mu}) \end{pmatrix} \begin{pmatrix} \rho_1(\boldsymbol{\mu}) & 0 \\ 0 & \rho_2(\boldsymbol{\mu}) \end{pmatrix} \begin{pmatrix} g_1(t) \\ g_2(t) \end{pmatrix}, \tag{3.1}$$

for $t \in [t_1, t_2]$. Each edge has to be either convex or concave. This corresponds to the definition of our curvy triangles in section 2.7.2. If we choose $g_1(t) = \cos t$ and $g_2(t) = \sin t$, we obtain the definition of an elliptical arc. A geometry defined in this way thus fulfills the affine geometry precondition.

The software first performs the three main steps described in section 2.7.3. We will now explain the first step - the generation of the RB triangulation on the reference domain Ω - in more detail. We recall that the aim is to construct a domain decomposition (2.123) of our reference geometry Ω and associated regions (defined by the user inputs) that is compatible with the interface condition (2.127). In addition, all elliptical and curvy triangles have to satisfy the consistency/continuity condition (2.139) and the internal angle conditions (2.147) and (2.149).

The software first focuses on all elliptical and curvy edges making part of the domain. For each elliptical or curvy arc, an elliptical, respectively curvy, triangle is introduced according to the definitions in section 2.7.2. In the case of internal interfaces, two triangles are needed. For each new triangle, an additional interior control point is added to the set of initial control points. If a triangle does not fulfill the internal angle conditions or the interface condition, this triangle is split by the software int two daughter triangles. This process is repeated until all introduced elliptical and curvy triangles are well-defined and consistent with the internal angle and interface conditions and the curved geometry is represented properly. After that, the algorithm fills the remainder of the domain with standard triangles by a Delaunay triangulation [68]. We note that although this algorithm guarantees a well-defined decomposition of the reference domain, it does not guarantee - even for parametric domains $\Omega_o(\boldsymbol{\mu})$ with only straight edge boundaries - that the Jacobians of the associated affine mappings will remain strictly positive for all $\boldsymbol{\mu} \in D$. This means that there may exist some parameter values $\boldsymbol{\mu}$ for which some of the triangles become singular and thus our domain decomposition is not "valid" anymore or (if the Jacobians do not vanish but get very small) the triangles are very distorted and lead to inefficient and ill-conditioned FE approximations of the underlying problem.

The parametrization of the curved geometry is unfortunately not arbitrary. The algorithm that determines the third point of the elliptical/curvy triangle requires that the origin $(O_1(\boldsymbol{\mu}), O_2(\boldsymbol{\mu}))^T$ of the curve in the form (3.1) must not lie on the segment of the curve. Furthermore, the choice of the origin has an influence on the quality of the generated triangles due to its influence on the location of the third point. In addition, we have another possibility to control the domain decomposition process by defining additional initial control points and edges that are not necessary for the specification of the geometry. As illustrated with several examples in [68], a good choice of the control points can reduce the number of affine terms Q_a by enforcing a domain decomposition which results in many parametrically similar subdomains. The proper selection of initial control points and edges can further ensure well-behaved/non-singular mappings. A bad choice of the control points can, in turn, lead to a very inefficient domain decomposition, when the decomposition algorithm is not able to find an affine decomposition that fulfills all requirements, where all interior

control points are situated in the domain boundaries and no subdomains intersect with any internal or external boundaries. The result is a huge number of subdomains, as the algorithm is trying to fulfill the internal angle conditions by performing more and more splits.

When the RB triangulation is established successfully, the underlying FE reference mesh is created based on this decomposition. The software provides the possibility to use either the meshing-facilities of the MATLAB PDE Toolbox® or to use the meshing routines of the COMSOL Multiphysics™package via the so-called "Comsol-interface". Other meshing tools and software can also be used, but then some changes to the source code have to be made. The meshes generated by the MATLAB routines are of relatively low quality as the routines do not offer many possibilities to control the meshing process. The COMSOL routines provide more possibilities to influence the generation and quality of the mesh. For the computations that have been made for this work, both variants have been used.

In the last steps, the parameter-dependent affine mappings for each subdomain are constructed and then translated into the PDE coefficients, as described in section 2.7.4. Finally, the FE system matrices and vectors are assembled (in the affine formulation (2.11)) and stored for the use by the low level RB kernel.

3.1.2 Problem Classes and Description

In the scalar elliptic case, the admissible class of problems is defined by (2.152) in the weak formulation. The original PDEs are then given by

$$-\frac{\partial}{\partial x_{oi}}\left(c_{ij}^k(\boldsymbol{\mu})\frac{\partial u}{\partial x_{oj}}\right)+U_i^k(\boldsymbol{\mu})\frac{\partial u}{\partial x_{oi}}+r^k(\boldsymbol{\mu})u=f^k(\boldsymbol{\mu}),\quad \forall\boldsymbol{\mu}\in D,\,1\le k\le K_{\mathrm{dom}} \tag{3.2}$$

in each subdomain k. The boundary conditions that can be imposed on each boundary edge can be Dirichlet conditions or Neumann/Robin conditions

$$n_{oi}c_{ij}^k(\boldsymbol{\mu})\frac{\partial u}{\partial x_{oj}}+h_1(u-h_2)=h_3,\quad \forall\boldsymbol{\mu}\in D,\,1\le k\le K_{\mathrm{dom}}. \tag{3.3}$$

The coefficients h_1, h_2 and h_3 may be functions of $\boldsymbol{x}$ and the parameter $\boldsymbol{\mu}$, and $\boldsymbol{n}$ is the outer normal to the domain boundary. Dirichlet conditions have to be constant values in the officially released version of the software package but can be dependent of $\boldsymbol{x}$ and $\boldsymbol{\mu}$ in an other, inofficial version. Both versions have been used for the computations in this work. As already mentioned before, inhomogeneous Neumann and Robin conditions are only allowed on straight and circular edges.

The user may also specify one (or more) outputs of interest. The software can handle the integral of the field over selected regions (defined by initial control points and edges) or over selected boundary edges and the integral of the flux, $n_{oi}c_{ij}^k(\boldsymbol{\mu})\partial u/\partial x_{oj}$ over selected boundary edges.

If the field variable is vector valued, the software provides a set of predefined problem types for linear elasticity problems. The general linear elasticity PDE solved by the program is given by

$$-\frac{\partial}{\partial x_{oi}}\left(c_{ijlm}^k(\boldsymbol{\mu})\frac{\partial u_l}{\partial x_{om}}\right)+r_i^k(\boldsymbol{\mu})\delta_{ij}u_j=b_i^k(\boldsymbol{\mu}),\quad r_i^k\ge 0,\quad \forall\boldsymbol{\mu}\in D,\,1\le k\le K_{\mathrm{dom}}, \tag{3.4}$$

for the displacement variable $\boldsymbol{u}=(u_1,\,u_2)^T$. Boundary conditions can again be of Dirichlet or Neumann/Robin type. The possible outputs are now the integral of components of the field variable over selected regions or over selected boundary edges and the integral of the stress $C_{ijlm}(\boldsymbol{\mu})(\partial u_l/\partial x_m)n_j$ over selected boundary edges.

In addition to the geometry and the problem to be solved, the user has to specify the parameter range D for the problem, the reference parameter $\boldsymbol{\mu}_{\mathrm{ref}}$ and the value for $\overline{\boldsymbol{\mu}}$ in the X-norm definition

(2.6). The parameter range has important implications on the performance of the RB method. First, a bigger parameter range will require more basis functions, although this dependency is relatively weak. More important is to keep in mind the implications of the parameter range on the geometrical variations. The geometry should be well-defined for all $\boldsymbol{\mu} \in D$, and also the domain decomposition must not become singular for any $\boldsymbol{\mu} \in D$. While the correctness of the different parts of the geometry over the whole parameter domain can be assured in advance by the user, the correctness of the affine decomposition is of course difficult to guarantee. An "intelligent" choice for the reference parameter and the parameter domain is an important approach to obtain a somewhat "robust" decomposition.

We will now give a short overview of the subsequent steps performed by the software in the offline and online phase.

3.1.3 Offline and Online Computation

In the offline stage, after the symbolic preprocessing steps, the software first builds the greedy "train" sample Θ_{train} and performs the SCM algorithm to compute the lower bounds for the coercivity constants for all members $\boldsymbol{\mu} \in \Theta_{\text{train}}$. After that, the primal reduced basis space is generated by the greedy algorithm. In the case of noncompliant problems, also the dual reduced basis space is generated by a second run of the greedy algorithm for the dual problem. The resulting primal and dual spaces are stored to be available in the online stage.

In the online stage, the solution and output for a particular parameter value $\boldsymbol{\mu}$ are computed. For the computation of the solution, the user can either prescribe a value for the desired accuracy or the number of basis functions that the software shall use. In the first case, the software determines the number of basis functions necessary to fulfill the desired accuracy and then computes the field solution based on the primal approximation space. In the latter case, the field solution is computed directly with the prescribed number of primal basis functions. In both cases, the lower bound for the coercivity constant for the parameter value $\boldsymbol{\mu}$ is computed in order to calculate the a posteriori error bound (2.46) for the field variable. After this, the field variable is used together with the dual reduced basis space to compute the output (or outputs, respectively). Also for the outputs the user can choose between the possibility to prescribe an output error tolerance or to prescribe the number of dual basis functions to be used. The output error estimate is then computed after (2.116). The usage of the symbolic preprocessor also allows a visualization of the field solution and the deformed mesh.

3.1.4 Standard Settings

In all computations of the subsequent chapter, the standard values of the rbMIT software package are used for the parameters controlling the RB offline and online computations. The most important settings are listed in table 3.1. The first three parameters thereby correspond to the greedy algorithm for the computation of the RB spaces, see section 2.3.5. ϵ_{tol} is the error tolerance in the greedy algorithm (2.28), $N_{\max}$ is the maximum number of basis function/greedy steps allowed in (2.28) and n_{train} is the number of elements in the greedy "train" sample Ξ_{train}. The remaining parameters in table 3.1 correspond to the SCM algorithm (2.92) in section 2.5.2. ϵ_{SCM} is the maximum error tolerance for α_{tol}, $J_{\max}$ and M define the coercivity constraint sample (2.81) and its subsets, while $n_{\text{train,SCM}}$ denotes the number of elements in the SCM "train" sample.

ϵ_{tol}	$N_{\max}$	n_{train}	ϵ_{SCM}	$J_{\max}$	M	$n_{\text{train,SCM}}$
0.01	120	1000	0.85	200	16	3000

Table 3.1: Standard settings of the rbMIT software package.

3.2 Geometric Setups

The computations in this work have been done for three different geometric problems in several variants. The geometries consist of basically rectangular domains with one or two centered airfoil profiles. We will now define these geometries in more detail.

3.2.1 The NACA 4-digits Family - Symmetric Case

The airfoil profiles we are considering in this work are symmetric and belong to the NACA 4-digits family. This allows us to work with geometries that are sufficiently complex and have some practical relevance but do not involve too many parameters. This leaves us some freedom to introduce other parameters (such as the angle of attack or relative locations of two profiles etc.). Profiles of the NACA 4-digits family are fully described by a thickness distribution and an equation specifying the mean camber line. The mean camber line is defined as the line which has the same distance to the upper side of the profile and to the lower side of the profile. In the non symmetric case a member of the NACA 4-digits family is described by three different parameters describing the possible variations in geometry. The first parameter is the maximum value of the mean-line ordinate (first integer in the numbering system), the second parameter is the distance from the leading edge to the location of the maximum camber (second integer in the numbering system) and the last parameter is the maximum thickness of the profile (last two integers of the numbering system). In the symmetric system, only the third parameter is relevant. It is given as a percentage of the length of the profile. Without a loss of generality, we always assume profiles of unity length in the following geometries. Of course, if the combination of two profiles with different lengths shall be investigated, another parameter for the relative length of the profiles to each other should be introduced.

The equation describing the thickness distribution is given by the following (see [1]):

$$\pm\, y_t = \frac{t}{0.2}\left(0.29690\sqrt{x} - 0.12600x - 0.35160x^2 + 0.28430x^3 - 0.10150x^4\right), \tag{3.5}$$

for the upper, respectively lower, side of the profile. In this case, t is defined as the maximum thickness as a fraction of the chord length (the length of the profile measured from the nose to the tail). As the coefficients do not sum up exactly to one, a slight modification has been introduced in the higher order terms to close the profile. With μ_1 being the maximum thickness as a percentage of the chord, the resulting equation is

$$\pm\, y_t = \frac{\mu_1}{20}\left(0.2969\sqrt{x} - 0.1260x - 0.3520x^2 + 0.2832x^3 - 0.1021x^4\right). \tag{3.6}$$

This parametrization can easily be transformed to the necessary form (3.1):

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & \pm\mu_1 \end{pmatrix} \begin{pmatrix} t \\ 0.2969\sqrt{t} - 0.1260t - 0.3520t^2 + 0.2832t^3 - 0.1021t^4 \end{pmatrix} \tag{3.7}$$

with $t \in [0, 1]$. There are two problems with this parametrization: First, the derivative of the x_2-coordinate is not defined for $t = 0$. Second, the origin in this formulation is $(0, 0)^T$ and is thus included in the curve itself. To avoid these problems, two different strategies have been chosen. On the one hand, t is replaced by t^2. Then the derivatives are well defined even for $t = 0$. Of course, the resulting equation is of 8^{th} order, what makes the problem more complicated and expensive for the (symbolic) domain decomposition algorithm. In practice, the software performed relatively well even for this complex parametrization. The second "trick" is to define the curve in different segments, while the first segment which includes the point $(0, 0)$ is defined relative to the origin $(1, 0)$, and the other segment which includes the point $(1, 0)$ is defined relative to the origin $(0, 0)$.

In addition, the first segment is split into several parts which has a positive influence on the affine domain decomposition and on the resulting FE-mesh. The resulting parametrization is:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ 0 & \pm\mu_1 \end{pmatrix} \begin{pmatrix} 1 - t^2 \\ 0.2969t - 0.1260t^2 - 0.3520t^4 + 0.2832t^6 - 0.1021t^8 \end{pmatrix} \tag{3.8}$$

for $t \in [0, \sqrt{0.1}, \sqrt{0.3}]$ and

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & \pm\mu_1 \end{pmatrix} \begin{pmatrix} t^2 \\ 0.2969t - 0.1260t^2 - 0.3520t^4 + 0.2832t^6 - 0.1021t^8 \end{pmatrix} \tag{3.9}$$

for $t \in [\sqrt{0.3}, 1]$.

3.2.2 Problem 1: Single Profile, no Angle of Attack

The geometry of the first problem is based on a rectangular domain with a single profile situated in the middle of the domain. The free stream with the velocity $\boldsymbol{u}_\infty$ (fixed) approaches the profile from the left side with zero angle of attack. This is the basis for two different variants of this problem and all following problems.

Variant A: No ground effect

The first variant of this problem considers a profile which is situated far from the ground, that means that no ground effect has to be included and the domain and boundary conditions are symmetrical to the x_1 axis. Hence, this variant is a single-parameter problem. The parameter is the thickness of the profile, μ_1 defined in percent of the chordlength L of the profile. A sketch of this setup is depicted in figure 3.1. The thickness parameter induces a relatively simple stretch of the profile in the x_2-direction and could thus also be modeled by stretching the whole domain, but with respect to the following, more complex setups, we did not follow this alternative approach.

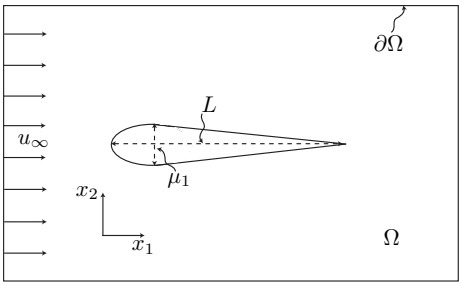


Figure 3.1: Setup of Problem 1, variant A: single profile, no angle of attack, no ground effect.

Variant B: Ground effect included

In the second variant, the profile is no longer situated far from the ground, that means that ground effects become important and a second parameter, the distance between the profile and the ground, has to be introduced. We now have a two-parameters problem with μ_1 being the thickness as defined for variant A and μ_2 being the minimum distance between the lower part of the profile and the ground. This setup is shown in figure 3.2.

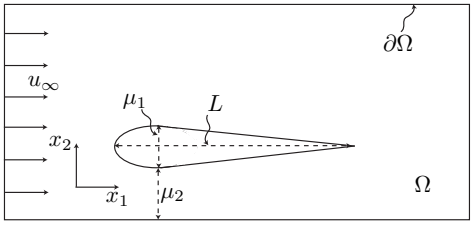


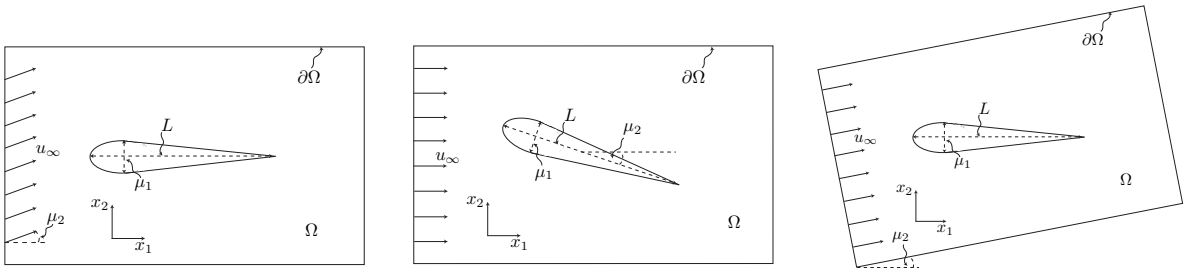
Figure 3.2: Setup of Problem 1, variant B: single profile, no angle of attack, ground effect included.

3.2.3 Problem 2: Single Profile, with Angle of Attack

The second problem is based on the first problem, but now the free stream approaches the profile from the left side with an angle of attack greater than zero. Again, we consider two different variants of the problem. In this case, there exist several possible realizations of the parametrized geometries with the tools that the symbolic preprocessor offers. We will briefly present them for variant A.

Variant A: No ground effect

This problem contains two parameters, the thickness μ_1, as defined before, and the angle of attack μ_2. The basic geometry in this variant is identical to problem 1, variant A, as depicted in figure 3.3(a). The angle of attack is here implemented in the boundary conditions for the free stream velocity u_∞. The profile and the domain boundaries remain fixed. The second realization is depicted in figure 3.3(b). Here, the profile is rotated according to the angle of attack, while the domain boundaries remain fixed and the free stream is parallel to the x_1-axis. In the third realization, shown in figure 3.3(c), the profile remains fixed but the domain boundaries are rotated. The free stream is parallel to the upper and lower domain boundaries and perpendicular to the left and right domain boundaries. The angle of attack has been chosen to be positive, as it is relevant for the design of wings for airplanes. Because in the domain of race car engineering negative angles of attack are needed and the profiles are symmetric, we can just reflect the setup with respect to the x_1 axis to obtain the relevant solutions and forces for race car design.



(a) (b) (c)

Figure 3.3: Setup of Problem 2, variant A: single profile, with angle of attack, different realizations.

Variant B: Ground effect included

The second variant of this problem again includes ground effects in the computation as the profile is situated close to the ground. This situation is depicted in figure 3.4. This problem involves three parameters. The thickness μ_1 and the angle of attack μ_2 are defined as for variant A, but in

addition we have to consider the distance between the profile and the ground μ_3. This parameter is now defined as the minimum distance between the lower side of the profile and the ground for zero angle of attack. We observe that the angle of attack is now defined to be negative. This is necessary as the presence of the ground makes it impossible to simply reflect the setup with respect to the x_1-axis as it is the case for variant A.

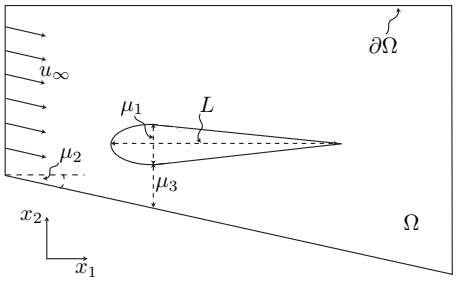


Figure 3.4: Setup of Problem 2, variant B: single profile, with angle of attack, ground effect included.

3.2.4 Problem 3: Two Profiles

The last problem includes two identical profiles. This setup is shown in figure 3.5. The two profiles are positioned far from the ground, so ground effects are not relevant. We now have 4 parameters involved in the setup: the thickness and angle of attack μ_1 and μ_2, respectively, and the new parameters that describe the distance of the profiles in the x_1-axis and in the x_2 axis, μ_3 and μ_4, respectively. The distance in x_1-direction is defined as the absolute distance between the centers of the profiles in x_1-direction, while the distance in x_2-direction is defined as the relative distance between the centers of the profiles in x_2-direction.

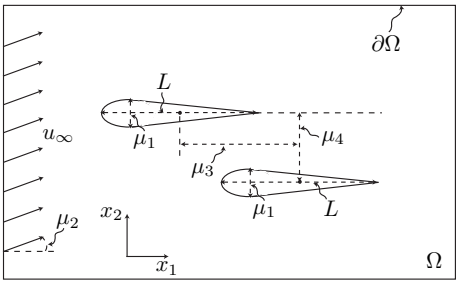


Figure 3.5: Setup of Problem 2: two profiles.

3.3 Reduced Order Modeling: Simplified Flow Models

This section is devoted to another possibility to achieve a reduction of the computational effort to solve a certain physical problem. There exists an approach which is much more common than reducing the size of the problem by "general" procedures such as the reduced basis method: The complexity of the problem is very often (in fact mostly) reduced by a simplification of the physical model which describes the problem. This approach is not general but the possible reductions have to be determined as the case arises. Based on the full description of all physics involved in the problem, the physical phenomena are reduced to those being really essential for this problem. In

many engineering applications, the reduced models obtained are absolutely sufficient to solve the problem with the required accuracy, but it demands a lot of experience to be able to decide what is important and what not.

In the field of fluid dynamics, there exist a lot of different simplified flow models. The range of models is very broad, as each model suits a particular class of applications, and the spectrum of applications in fluid dynamics is huge. Examples for such fluid models used in this work are potential flows, convection-diffusion models and creeping flows described by the Stokes system. One of the most complex models used in fluid dynamics are the Navier-Stokes equations, but even here, we presume certain qualities of the fluid, as it has to behave Newtonian, and we thus simplify the physics involved in the real problem. Nevertheless, the solution of the Navier-Stokes equations with an extremely fine mesh resolution for fluids that are indeed of the Newtonian type, can be considered to be very close to reality.

The simplification of the solution results from two different effects. First, the concentration on the essential physical phenomena can cause a less complex mathematical model. That means that the resulting system of partial differential equations describing the problem is easier and cheaper to solve numerically. For example, the Stokes equations are a simplification of the Navier-Stokes equations that do not contain any nonlinear convective acceleration terms as the inertial forces can be neglected for creeping flows. This makes the solution process a lot simpler. The second effect is that the simplification of the physical model also causes a simplification in the scales involved in the problem. If for examples the small scale turbulent variations are considered to be negligible (for example in the model of Potential flows), the computation can be done on a much coarser grid and thus much computational time can be saved.

But we also have to keep in mind the possible disadvantages of such simplified models. If we simplify the model too much, significant effects can be excluded of the computations and the solution behavior can be changed dramatically. As already mentioned, it requires a lot of experience to evaluate which simplifications can be made for a particular problem and which would have a severe effect on the correctness and accuracy of the solution. Very often, the simplifications are only fully valid for some special cases with very high accuracy, but for other applications the accuracy obtained can be still good enough, although the modeling assumptions are not fully met anymore. Sometimes this is a balancing act between physical correctness and computational savings, respectively feasibility.

A common approach is to consider models of incrementally increasing complexity. This allows us to gradually increase the understanding of the fundamental effects involved in a problem and to decide in which regions of a flow field a more detailed model is indeed necessary and in which regions the simpler model is absolutely sufficient. This helps us to keep the accuracy high while still benefiting from the cheaper simplified models.

The computations in this work also follow the incremental approach. We first applied the reduced basis method to the model of Potential flows, which results in the solution of a simple linear symmetric coercive scalar system, the Laplace equation (see chapter 4). The next step was the solution of a convection-diffusion scalar model by the reduced basis method. The convection-diffusion system (see chapter 5) includes an additional convection term. This means that the resulting system is no longer symmetric and an additional scale is introduced which has to be resolved. The last level of complexity considered in this wok are creeping flows represented by the Stokes system. This system is now vectorial and includes an additional variable, the pressure. In addition, it is no longer coercive. To recover coercivity and eliminate the pressure as unknown in the equation system, we used a penalty formulation of the Stokes system (see chapter 6) and thus sacrifice the compatibility with the continuity equation. The ultimate goal would be the application of the RB method to the full Stokes system (see chapter 7) and finally to the Navier-Stokes equations, but this would go beyond the scope of this work. The detailed implications of the modeling assumptions on the representation of the physics involved in the problems and on

Model	Modeling assumptions	System properties
Potential flows	- non-viscous flows - velocity field as gradient of a scalar function - irrotational velocity field - no boundary layer - laminar flow - no lift/drag without circulation	- scalar - coercive - symmetric - linear
Convection-diffusion	- transport of a scalar in a given flow field - convective and diffusive effects - no feedback of the scalar to the flow field.	- scalar - coercive - **nonsymmetric** - linear
Stokes (penalty)	- creeping flows: inertial forces neglected - no convective terms - highly viscous flows or slow motion or small dimensions of domain or objects - laminar flow - boundary layer - aerodynamic forces appear - equations for the velocity of the flow field - continuity equation not fulfilled exactly $\rightarrow$ mass sinks and sources appear	- **vectorial** ($\nu = d$) - coercive - symmetric - linear
Stokes (full system)	- same qualities as above but - continuity equation fulfilled - equations for the velocity and pressure	- vectorial ($\boldsymbol{\nu = d + 1}$) - **noncoercive** - nonsymmetric - linear
Navier-Stokes	- inertial forces important - Newtonian fluid - possibly turbulent	- vectorial ($\nu = d + 1$) - noncoercive - nonsymmetric - **nonlinear**

Table 3.2: Summary of simplified flow models with increasing complexity

the solution process are discussed in the subsequent chapters, but we will give a short summary of modeling assumptions and computational benefits in table 3.2. All models are applied in the stationary variant for incompressible flows. The implications of these assumptions are therefore not considered in the modeling assumptions.

4 The Reduced Basis Method for Potential Flows around an Airfoil

In chapter 2.3 we presented the reduced basis method for the approximation of the solution of parametrized elliptic coercive partial differential equations. Rigorous, efficient and sharp a posteriori error bounds have been developed in section 2.4. We further introduced in section 2.7 the concept of affine geometric variations and affine geometry decompositions.

In this chapter, we now apply the reduced basis approximation and error bounds together with the affine geometry decomposition to potential flows. The basic geometries on which we want to obtain a solution for the flow field have been presented in section 3.2: We consider the flow around one or two airfoils in a rectangular domain. The potential Flows can be considered as one of the simplest two- or three-dimensional flow models which result in a partial differential equation from the mathematical point of view.

In the following sections, we will first precise the definition and properties of potential flows. We will then reformulate the necessary equations in order to obtain a problem equivalent to (2.1). As the special structure of potential flows allows us to solve a scalar equation for the so-called potential function instead of a vectorial system for the velocity components and the pressure, there is also a new set of special a posteriori error bounds for velocity and pressure, based on the original error bounds for the scalar variable, the potential function. Finally, the numerical results for the different geometrical setups that have been obtained by the use of the rbMIT software package (see chapter 3.1) will be presented.

4.1 Potential Flows

Potential flows describe laminar non-viscous and irrotational flows [9, 53, 71]. Irrotational velocity fields can be defined as the gradient of a so-called potential function ϕ:

$$\boldsymbol{u} = \nabla\phi, \tag{4.1}$$

where $\boldsymbol{u} = (u_1, u_2)^T$ is the velocity with the velocity components in x_1 and x_2 direction. Together with the steady continuity equation $\nabla \cdot \boldsymbol{u} = 0$, we obtain the governing Laplace-equation for the potential:

$$-\Delta\phi = 0 \quad \text{in } \Omega_o(\boldsymbol{\mu}). \tag{4.2}$$

The pressure p at an arbitrary point in the domain $\Omega_o(\boldsymbol{\mu})$ can then subsequently be calculated with the help of Bernoulli's equation:

$$p + \frac{1}{2}\rho|\boldsymbol{u}|^2 = p_{\text{in}} + \frac{1}{2}\rho|\boldsymbol{u}_{\text{in}}|^2, \quad \text{in } \Omega_o(\boldsymbol{\mu}), \tag{4.3}$$

and the pressure coefficient c_p can be defined as

$$c_p = \frac{p - p_{\text{in}}}{\frac{1}{2}\rho|\boldsymbol{u}_{\text{in}}|^2} = 1 - \left(\frac{|\boldsymbol{u}|^2}{|\boldsymbol{u}_{\text{in}}|^2}\right). \tag{4.4}$$

In these equations, p_{in} is the pressure of the undisturbed flow on the inflow boundary and $\boldsymbol{u}_{\text{in}}$ is the velocity vector of the undisturbed flow on the inflow boundary. Gravity effects are not included in this formulation, but could easily be added to the equation. Furthermore, a time dependency could be introduced in the pressure calculation by using a time dependent formulation of Bernoulli's equation, see e.g. [71]. Boundary conditions are given by homogeneous Neumann conditions

$$\frac{\partial \phi}{\partial n} = 0 \quad \text{on } \Gamma_{\text{w}}(\boldsymbol{\mu}) \tag{4.5}$$

to describe non-penetration on walls $\Gamma_{\text{w}}(\boldsymbol{\mu})$, inhomogeneous Neumann conditions

$$\frac{\partial \phi}{\partial n} = u_{\text{in}} \quad \text{on } \Gamma_{\text{in}}(\boldsymbol{\mu}) \tag{4.6}$$

to impose the inflow velocity on the inflow boundary $\Gamma_{\text{in}}(\boldsymbol{\mu})$ and by (homogeneous or inhomogeneous) Dirichlet conditions

$$\phi = \phi_{\text{ref}} \quad \text{on } \Gamma_{\text{out}}(\boldsymbol{\mu}) \tag{4.7}$$

to prescribe the level of the potential on the outflow boundary $\Gamma_{\text{out}}(\boldsymbol{\mu})$.

As these idealized flows do not include any viscous effects, they have a velocity different from zero on solid walls, while realistic viscous flows meet the no-slip condition. If the complete flow field is modeled by potential flows, we can not compute any aerodynamic forces such as lift and drag unless we introduce artificial circulation to the flow field. One reason for this is the nonviscous character of the flow field - the flow can not exert any viscous forces on the surface of an object. In addition, potential flows are able to flow around sharp corners without any separation effects due to the absence of viscosity. Together with the requirement of an irrotational flow field, this leads to a flow field that exerts neither drag nor lift forces on any object placed in the flow field. Concerning the generation of lift, the introduction of artificial circulation could partly cure this problem as it can enforce a "correct" flow field, where the so-called Kutta-condition is met [9]. Explained by means of an airfoil, the Kutta-condition requires that the rear stagnation point is situated on the sharp trailing edge and not on the upper or lower side of the profile (as it is the case for potential flows without artificial circulation). The flow field can then generate lift. The artificial circulation could be introduced through the boundary conditions, but would have to be adjusted by a "try and error" method. Therefore we did not follow this approach but decided to omit the computation of the aerodynamic forces for this flow model. Instead, we concentrated on the computation of the entire velocity and pressure fields.

4.2 Problem Statement

We now derive the weak formulation of the governing equations. To begin, we recall the function space X^e and associated norms defined in section 2.2.1. Multiplying (4.2) by a test function $v \in X^e$ and integrating over $\Omega_o(\boldsymbol{\mu})$, we obtain

$$-\int_{\Omega_o(\boldsymbol{\mu})} v \nabla^2 \phi = 0, \quad \forall v \in X^e. \tag{4.8}$$

Integrating by parts and applying the divergence theorem yields

$$-\int_{\Omega_o(\boldsymbol{\mu})} v \nabla^2 \phi = \int_{\Omega_o(\boldsymbol{\mu}} \nabla \phi \cdot \nabla v - \int_{\Omega_o(\boldsymbol{\mu})} \nabla \cdot (v \nabla \phi) \tag{4.9}$$

$$= \int_{\Omega_o(\boldsymbol{\mu})} \nabla \phi \cdot \nabla v - \int_{\partial \Omega_o^N(\boldsymbol{\mu})} v \frac{\partial \phi}{\partial n} \quad = 0, \qquad \forall v \in X^e, \tag{4.10}$$

where $\partial\Omega_o^N(\boldsymbol{\mu})$ is the part of the boundary on which inhomogeneous Neumann conditions are prescribed. From this, we obtain our weak statement as

$$a(w, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in X^e, \tag{4.11}$$

with

$$a(w, v; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} \nabla w \cdot \nabla v \tag{4.12}$$

$$f(v; \boldsymbol{\mu}) = \int_{\partial\Omega_o^N(\boldsymbol{\mu})} v \frac{\partial w}{\partial n}. \tag{4.13}$$

From the formulation (4.11) on the parameter dependent domain $\Omega_o(\boldsymbol{\mu})$, the formulation on the reference domain Ω can be derived with the methods introduced in section 2.7. The main steps are the affine decomposition of the original domain, the identification of the affine mappings and in the end the "translation" of the affine mapping coefficients to the formulation on the reference domain (2.1).

4.3 Error Bounds for Velocity and Pressure

The a posteriori error bounds developed in section 2.4 can be used to bound the error in the solution for the scalar potential function. To get an error bound for the error in velocity and pressure, we have to modify this bound slightly. The following development is described in detail in [54]. We start with a given triangulation $\mathcal{T}_{N_t}$ of the domain with triangles T_{N_t}. The kinetic energy on a triangle T_{N_t} can then be represented by

$$K_{N_t}^{T_{N_t}} = (\nabla\phi^{N_t})^2 = \frac{1}{|T_{N_t}|} \int_{T_{N_t}} |\nabla\phi^{N_t}|^2 \quad \text{in } T_{N_t}. \tag{4.14}$$

The potential function ϕ can be expected to be smooth enough to give $\nabla\phi \in C^0(\Omega)$ and $\nabla\phi$ can be interpreted pointwise. $\phi^{N_t} \in X^{N_t}$ is the field solution of the FE "truth" approximation (2.2). The reduced basis solution $\phi_N^{N_t} \in W_N^{N_t} \in X^{N_t}$ then fulfills

$$a(\phi_N^{N_t}, v) = l(v), \quad \forall v \in W_N^{N_t}. \tag{4.15}$$

For simplicity, we omit the $\boldsymbol{\mu}$-dependency of the bilinear/linear forms and the error bounds in this section. Recall that the error between the reduced basis solution and the FE approximation $e_N^{N_t}$ is defined as

$$e_N^{N_t} = \phi^{N_t} - \phi_N^{N_t}. \tag{4.16}$$

With Δ_N^{en} being our usual energy error bound (2.46), we note that

$$a(e_N^{N_t}, e_N^{N_t}) = \int_\Omega |\nabla e_N^{N_t}|^2 \leq (\Delta_N^{\text{en}})^2 \tag{4.17}$$

follows from the inequalities in (2.50).

The RB approximation of $K_{N_t}^{T_{N_t}}$ is given as

$$K_N^{T_{N_t}} = (\nabla\phi_N^{N_t})^2 = \frac{1}{|T_{N_t}|} \int_{T_{N_t}} |\nabla\phi_N^{N_t}|^2 \quad \text{in } T_{N_t}. \tag{4.18}$$

We next introduce

$$I_N^{N_t} \equiv \left(\int_\Omega |\nabla \phi_N^{N_t}|^2 \right)^{1/2}. \tag{4.19}$$

Our aim is now to develop an error bound for the kinetic energy $K_N^{T_{N_t}}$. On a single triangle, we note that

$$\begin{aligned} K_{N_t}^{T_{N_t}} - K_N^{T_{N_t}} &= \left| \frac{1}{|T_{N_t}|} \int_{T_{N_t}} |\nabla \phi^{N_t}|^2 - |\nabla \phi_N^{N_t}|^2 \right| \\ &= \frac{1}{|T_{N_t}|} \int_{T_{N_t}} \nabla(\phi^{N_t} - \phi_N^{N_t}) \cdot \nabla(\phi^{N_t} + \phi_N^{N_t}) \\ &= \frac{1}{|T_{N_t}|} \int_{T_{N_t}} \nabla(\phi^{N_t} - \phi_N^{N_t}) \cdot \nabla(\phi^{N_t} - \phi_N^{N_t}) + 2\nabla(\phi^{N_t} - \phi_N^{N_t}) \cdot \nabla \phi_N^{N_t}. \end{aligned} \tag{4.20}$$

Hence

$$\begin{aligned} &\sum_{T_{N_t} \in \mathcal{T}_{N_t}} |T_{N_t}| |K_{N_t}^{T_{N_t}} - K_N^{T_{N_t}}| \\ \leq &\sum_{T_{N_t} \in \mathcal{T}_{N_t}} \int_{T_{N_t}} \nabla(\phi^{N_t} - \phi_N^{N_t}) \cdot \nabla(\phi^{N_t} - \phi_N^{N_t}) + 2 \sum_{T_{N_t} \in \mathcal{T}_{N_t}} \int_{T_{N_t}} \nabla(\phi^{N_t} - \phi_N^{N_t}) \cdot \nabla \phi_N^{N_t} \\ \leq &(\Delta_N^{\text{en}})^2 + 2 I_N^{N_t} \Delta_N^{\text{en}} \equiv \Delta_N^K. \end{aligned} \tag{4.21}$$

Δ_N^K is a $L^1(\Omega)$ error bound for the RB prediction for the squared velocity ($|\nabla \phi_N^{N_t}|^2$).

An error bound for the pressure can be obtained using (4.21) and Bernoulli's equation. If we define

$$p_{N_t}^{T_{N_t}} = \frac{1}{|T_{N_t}|} \int_{T_{N_t}} (B_{\text{in}} - \frac{\rho}{2} |\nabla \phi^{N_t}|^2), \tag{4.22}$$

$$p_N^{T_{N_t}} = \frac{1}{|T_{N_t}|} \int_{T_{N_t}} (B_{\text{in}} - \frac{\rho}{2} |\nabla \phi_N^{N_t}|^2), \tag{4.23}$$

where $B_{\text{in}} \equiv p_{\text{in}} + \frac{\rho}{2} |\boldsymbol{u}_{\text{in}}|^2$ is given, the error bound for the pressure follows directly as

$$\sum_{T_{N_t} \in \mathcal{T}_{N_t}} |T_{N_T}| |p_{N_t}^{T_{N_t}} - p_N^{T_{N_t}}| \leq \frac{\rho}{2} \Delta_N^K \equiv \Delta_N^p. \tag{4.24}$$

We note that we can also develop similar error bounds for the pointwise squared velocity $|(\nabla \phi_o^{N_t})^2 - (\nabla \phi_{N,o}^{N_t})^2|$, see [54].

4.4 Numerical Results

We applied the RB method as implemented in the rbMIT software package to potential flows in the different geometric setups presented in section 3.2. The emphasis thereby has been laid on the affine decomposition of the reference domains, the (offline) generation of suitable reduced basis spaces for the approximation of the field solution and finally the online reduced basis computation of the field solution and the associated a posteriori error bounds. The expression "field solution" shall in the context of potential flows not only apply to the potential function, but also to the velocity and pressure distributions. Tables 4.1 and 4.2 provide an overview of the different geometric setups together with the special issues treated for each setup. In the following sections, we will then

	P1VA	P1VB	P2VA	P2VB	P3
# profiles	1	1	1	1	2
angle of attack	no	no	yes	yes	yes
ground effect	no	yes	no	yes	no

Table 4.1: Overview of the geometric setups

Problem	Topics
P1VA	- affine geometry decomposition (for 3 different setups) - FE mesh and mesh deformation - field solution - comparison real error/estimated error
P1VB	- SCM algorithm (convergence of upper and lower bounds) - distribution of the greedy sample points - field solution - comparison real error/estimated error and error for different N
P2VA	- distribution of the greedy sample points - field solution
P2VB	- field solution - mesh deformation - comparison real error/estimated error and error for different N
P3	- field solution - RB online time as a function of N

Table 4.2: Overview of the topics treated for each setup.

subsequently address the different geometric setups and present the relevant numerical results. The greedy algorithm is based on an estimate for the relative error instead of an estimate for the absolute error.

4.4.1 Problem 1: Single Profile, no Angle of Attack

Variant A: No Ground Effect

The rectangular basic domain in which the airfoil is positioned is $[-6, 6] \times [-4, 4]$. In the following, we will refer to this basic domain as the "big domain" in contrast to a domain which stretches over $[-1, 3] \times [-2, 2]$, which we will call "small domain". As already mentioned, this setup represents the most simple configuration we worked with, as it involves only a single parameter, the thickness μ_1. In this variant, the parameter may vary in $\mu_1 \in [4, 24]$, while the reference parameter μ_{ref} and energy-norm parameter $\overline{\mu}$ are chosen to be $\mu_{\text{ref}} = \overline{\mu} = 14$. The variations of the geometry introduced by the thickness parameter are relatively modest and thus the parameter range is not critical and could also be chosen even larger. The boundary conditions are given by (4.5) (non-penetration) on the surface of the profile and the upper and lower domain boundary, while we use (4.6) with $u_{\text{in}} = 1$ (prescribed inflow velocity) on the left domain boundary and (4.7) with $\phi_{\text{ref}} = 0$ (parallel outflow) on the right domain boundary.

We first concentrate on the affine domain decomposition of the reference domain. The resulting triangulation is shown in figure 4.1. The domain decomposition algorithm creates 42 subdomains. In the nose region of the profile, where the curvature is high, the triangles are relatively small, while in the regions further away from the profile, the triangles are bigger. Some of them are relatively long stretched which can have a negative influence on the subsequent meshing procedure and on the conditioning of the resulting affine formulation of the bilinear form (2.11). Apart from this, the decomposition is very efficient, as it is symmetric with respect to the x_1 axis and many of the triangles are similar. To gain some control on the decomposition process, we introduced

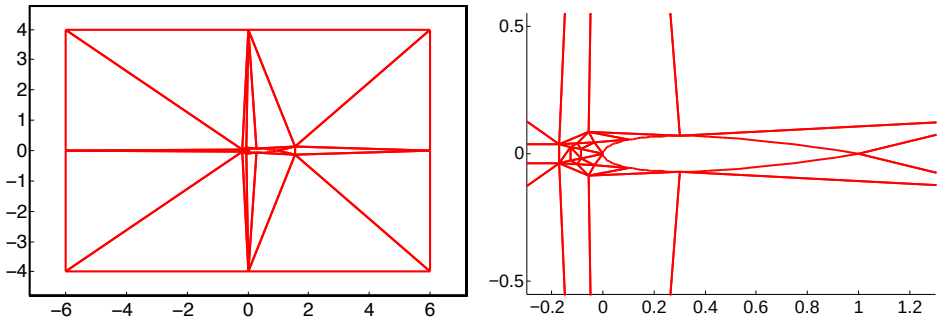


Figure 4.1: Final geometry decomposition, problem 1, variant A.

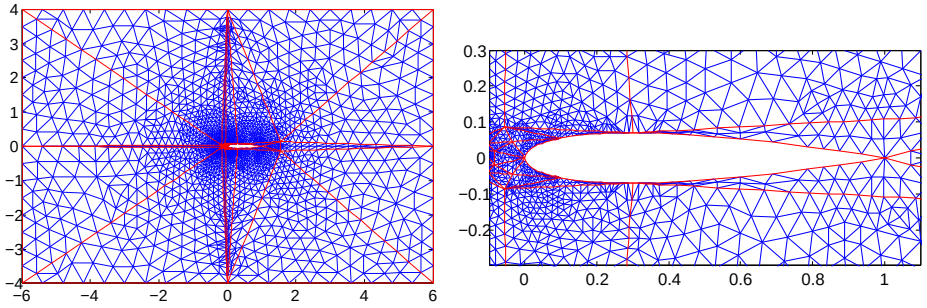


Figure 4.2: FE mesh with 3163 elements for problem 1, variant A.

four additional points on in the middle of the domain boundaries which enforce this advantageous decomposition. The efficiency of the domain decomposition is confirmed by the affine representation of the bilinear form (2.11) for this decomposition, which reduces the 42 subdomains to $Q_a = 16$ affine terms.

The last step of the symbolic preprocessing, the FE mesh generation, is illustrated in figure 4.2. The mesh used for this variant consists of 3163 triangular elements and has been generated by the Matlab subdivision method. The drawback of the low quality long-stretched triangles is obvious: The mesh quality in the regions with such acute-angled subdomains is very low. However, the mesh in the vicinity of the profile is fully sufficient for our purposes and potential flows are quite "robust" concerning mesh defects far from the airfoil. For other fluid models, other mesh generation procedures will be taken into account.

The SCM algorithm converges very fast as the required accuracy ϵ_{SCM} is fulfilled already after two iterations. Due to the relatively small variations in the solution caused by the parameter, only two RB basis functions are needed to satisfy the accuracy ϵ_{tol} in the greedy algorithm. The pressure field and the streamlines together with the velocity magnitude calculated by this reduced basis for the two extremal cases ($\mu_1 = 4$ and $\mu_1 = 24$) are shown in figure 4.3. The characteristic properties of potential flows can directly be recognized: The flow field is symmetric with respect to the x_2-axis, the velocity is not zero on the surface of the airfoils and there exist two stagnation points on the nose and the tail of the airfoils. Note that especially the tail stagnation point does not exhibit zero velocity as the mesh resolution is not sufficient to reproduce this effect properly. In addition, we note that potential flows are ideal flows as there occur no energy losses: The pressure level is the same on the inflow boundary and on the outflow boundary. The velocity on inflow and outflow boundary is equal as well, as this is required by the continuity equation and the assumption of a divergence free velocity field. To complete the presentation of the flow phenomena, the distribution of the pressure coefficient c_p around the profile is shown in figure 4.4, again for the two extremal cases. The symmetry of the solution is almost perfect, apart from slight differences between the

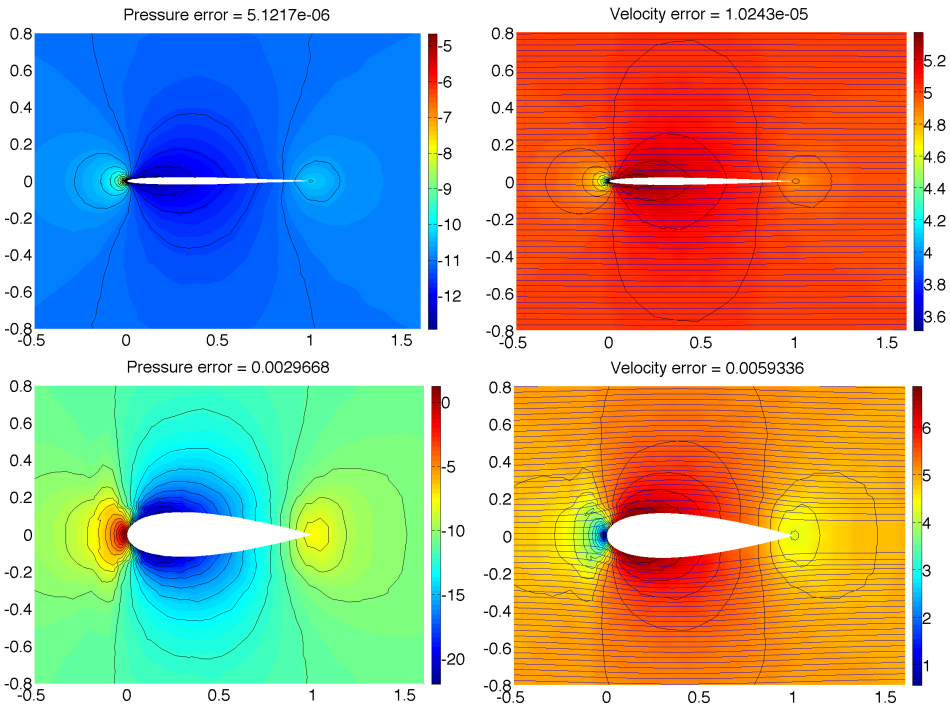


Figure 4.3: Pressure distribution and streamlines/velocity magnitude for a thickness of 4% (upper) and 24% (lower).

upper and the lower edge in the nose region due to mesh asymmetries. We note that the pressure on the surface of the thicker profile is lower than the pressure on the thin profile because, as a result of mass conservation, the surface velocities have to be higher in the case of a thicker profile.

The relation of the true error between the FE solution and the RB solution to the RB error estimate is illustrated in figure 4.5. Here, the true error in the energy norm of the kinetic energy is shown shown together with the estimated error of the kinetic energy (see section 4.3) in the energy norm over the whole parameter domain. It can be observed that the error estimate is indeed very sharp. The minimum errors (both in the real error and in the estimated error) naturally occur close to the two parameter sample points chosen by the greedy algorithm to generate the snapshots for the reduced basis. Far from these points, the error increases.

Figure 4.6 compares the meshes in the region close to the profile for the two extremal parameter instances $\mu_1 = 4$ and $\mu_1 = 24$. Here, the way in which the mesh deforms due to the affine transformations becomes visible and possible difficulties can be identified. The strongest deformation can be observed in a horizontal region as thick as the profile. This region is stretched or compressed to match the current parameter instance, while the outer regions remain almost unchanged (due to their bigger dimensions). The triangulation in the nose region becomes thus very narrow for small parameters while the size of the elements remains almost constant in the adjacent regions above and below the profile. In this case it is still uncritical, but for more complicated geometries this can lead to problems and the effects of parameter changes on the mesh should be observed to guarantee a good underlying FE solution.

To improve the quality of the affine geometry decomposition and the mesh quality, we designed two additional variants of this setup. In the first one, the rectangular domain has been shrunken to the size of the "small domain". This increases the minimum angle of the resulting triangles as they do not have to be as "long-stretched" anymore. In addition, more control points have been introduced on the domain boundary. The second variant embeds this small domain in the big domain in specifying the necessary control points and edges for both the small and the big domain.

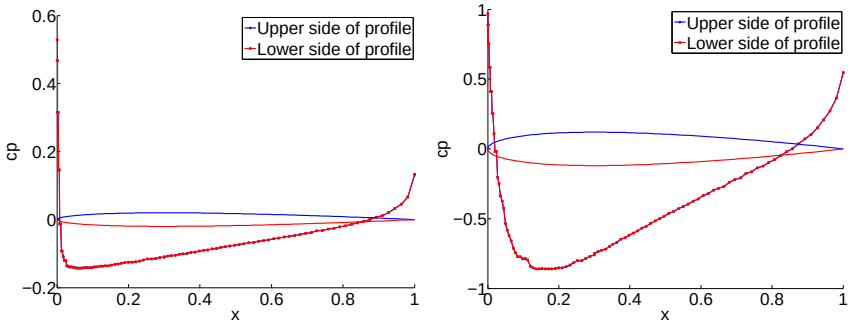


Figure 4.4: Distribution of c_p around the profile for $\mu_1 = 4$ (left) and $\mu_1 = 24$ (right).

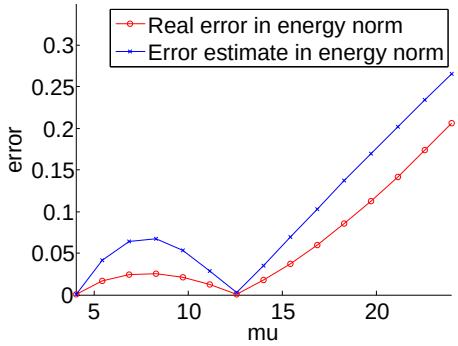


Figure 4.5: Real error and RB error estimate of the error in the kinetic energy (energy norm).

This approach exploits the advantages of the domain decomposition of a smaller domain, but leaves the overall dimensions of the domain unchanged. The results of the affine domain decomposition and the FE meshes for both variants can be found in appendix A.

The most important results of the offline RB generation process as well as the offline and online computational times are summarized in table 4.4. The offline time for the FE method t_{FE}^{offline} contains the times for the affine decomposition, the calculation of the (optimized) affine terms and the assembly of the FE system matrices. The offline time for the RB method t_{RB}^{offline} contains the FE offline time and the time necessary to generate the reduced basis function space (the time to perform the SCM algorithm for the coercivity lower bounds and the time to calculate the suitable basis functions by the greedy algorithm). We note that the FE offline time depends on the one hand on the number of the FE mesh elements N_t and on the other hand on the number of subdomains generated in the affine decomposition. The RB offline time then depends also on N_t and in addition on Q_a instead of the number of subdomains in the affine decomposition, as expected.

Even more important is of course the difference in the online performance between the FE and

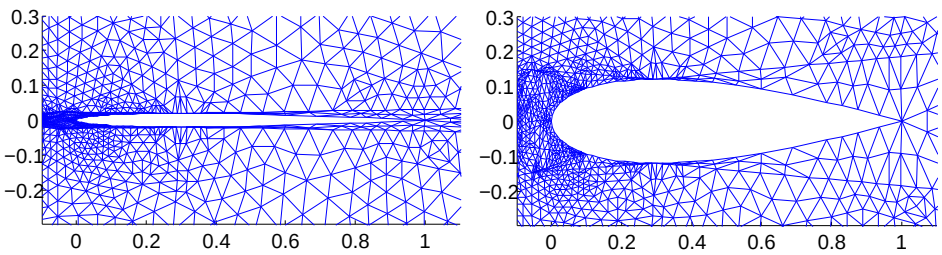


Figure 4.6: Details of the meshes for $\mu_1 = 4$ (left) and $\mu_1 = 24$ (right).

Symbol	Name	Explanation
t_{FE}^{offline}	FE offline time	affine decomposition + calculation of affine terms + assembly of FE system matrices
t_{RB}^{offline}	RB offline time	FE offline time + SCM algorithm for $\alpha_{LB}(\boldsymbol{\mu})$ + greedy algorithm for RB space
$t_{FE}^{\text{online},100}$	FE online time	computation of the field solution for 100 different parameter values (FE method)
$t_{RB}^{\text{online},100}$	RB online time	computation of the field solution for 100 different parameter values (RB method)
speedup	speedup	$t_{FE}^{\text{online},100}/t_{RB}^{\text{online},100}$

Table 4.3: The different computational times measured.

	Big Domain	Small Domain	Embedded Domain
# elements mesh N_t	3163	18771	5613
# subdomains	42	74	84
affine terms Q_a	16	20	16
affine terms Q_f	1	1	1
# RB functions $N_{\max}$	2	3	2
steps SCM $J_{\max}$	2	2	2
t_{FE}^{offline}	9min	58min	52min
t_{RB}^{offline}	19min	1h 44min	1h 2min
$t_{FE}^{\text{online, 100}}$	3.56s	25.72s	6.26s
$t_{RB}^{\text{online},100}$	$5.07{\cdot}10^{-3}$s	$7.21{\cdot}10^{-3}$s	$5.07{\cdot}10^{-3}$s
speedup	702	3567	1235

Table 4.4: Summary of the setups with zero angle of attack, no ground effect.

the RB method. We therefore compared the time for both methods for the online computation of the field solution for 100 different parameter instances, $t_{FE}^{\text{online},100}$ and $t_{RB}^{\text{online},100}$, respectively. The speedup is defined as the ratio of FE online time to RB online time. As expected, the RB method performs extremely well for all setups, as the number of basis functions $N = 2$ is indeed extremely low. Furthermore, the computations confirm that the speedup increases for a finer mesh, as desired. Finally, we note that the RB online times for the big domain and the embedded domain are equal, which confirms that the dominating variables for the online performance are Q_a and N (which are equal for both setups). The online computational time for the small domain is slightly higher which is caused by the higher number of basis functions N and affine terms in a Q_a. The different measures for the computational times introduced in this paragraph and their formation are summarized in table 4.3.

Variant B: Ground Effect Included

In this variant, the distance between the chord line of the airfoil and the ground in terms of the thickness μ_1 is introduced as a second parameter μ_2. The x_2-coordinate of the lower domain boundary is then defined as $x_2^l = -\mu_2 \cdot \mu_1/200$. The parameter range and reference parameters are indicated in table 4.5. The boundary conditions are specified according to variant A. Note that the parameter range has to be chosen carefully in order to avoid strong mesh distortion and to guarantee a correct underlying FE solution. The reference parameter also has a special importance in ensuring a well-defined affine mapping for all parameters. In this case, we avoided the formation of overlapping subdomains in the very narrow region between the profile and the ground for other

D_1	D_2	$\boldsymbol{\mu}_{\text{ref}}$	$\overline{\boldsymbol{\mu}}$
$[4, 24]$	$[1.5, 3]$	(14, 1.5)	(14, 2)

Table 4.5: Parameter range and reference parameters for problem 1, variant B.

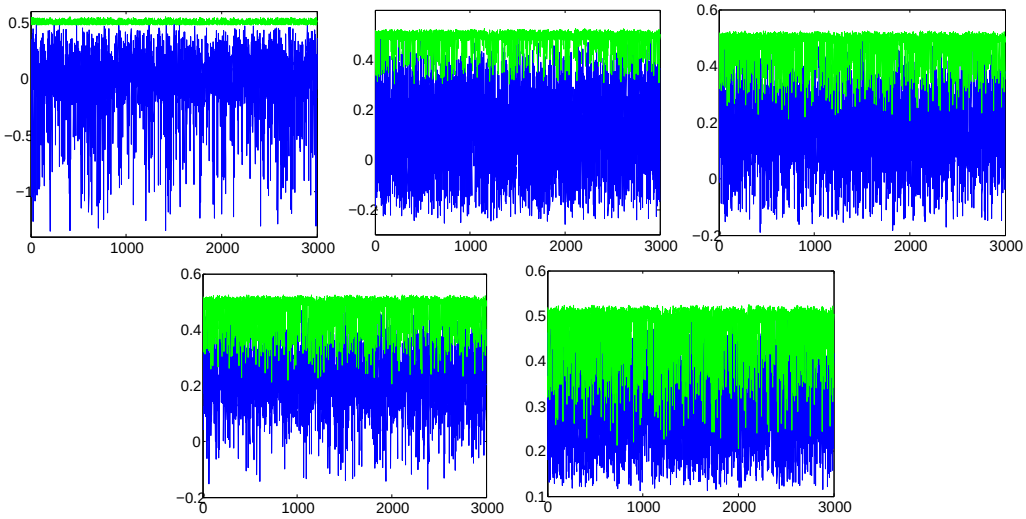


Figure 4.7: Upper (green) and lower (blue) bounds for the coercivity constant over the SCM parameter samples for each SCM step.

parameters than the reference parameter $\boldsymbol{\mu} \neq \boldsymbol{\mu}_{\text{ref}}$ by selecting the smallest distance as reference value. The resulting domain decompositions as well as the FE meshes can be found in appendix A. If not indicated differently, the following discussion concentrates on the results for the big domain.

The SCM algorithm needs 5 steps to calculate lower bounds for the coercivity constant α_{LB} which fulfill the accuracy requirements. The convergence behavior of the SCM algorithm is illustrated in figure 4.7. For each step, the distributions of the upper and lower bounds over the train sample $\Xi_{\text{train,SCM}}$ are depicted. In each step, the lower bounds approach the upper bounds. It becomes clear that the lower bounds can be negative in the beginning of the process, causing large values for α_{tol} in (2.92), but after some steps the lower bounds are positive for all parameter samples. However, the distribution of the lower bounds (and of course the "true" coercivity constants $\alpha(\boldsymbol{\mu})$) is not constant and differs a lot for the parameter samples. Bad numerical conditions can lead to situations, where convergence can be achieved for most parameter samples in the set $\Xi_{\text{train,SCM}}$, but

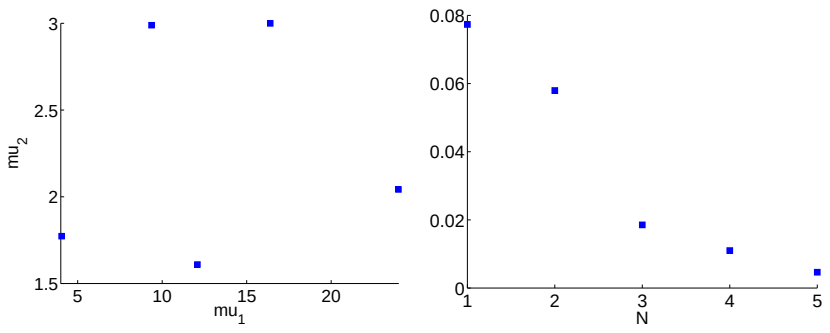


Figure 4.8: Sample points of the greedy RB space and maximum relative energy error as a function of the number of basis functions N for problem 1, variant B.

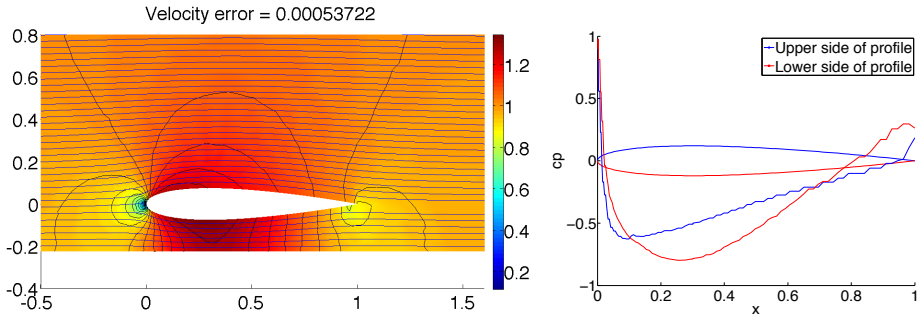


Figure 4.9: Streamlines with velocity magnitude and distribution of cp around the profile for a thickness of 15% and a relative distance of 1.5.

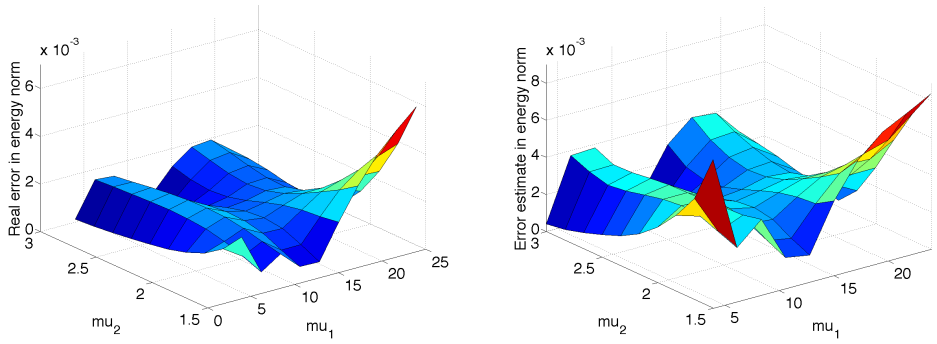


Figure 4.10: Real error and RB error estimate of the error in the kinetic energy (energy norm) for problem 1, variant B.

for one or a few single parameter samples no suitable lower bound for the coercivity constant can be calculated. As the greedy algorithm is able to generate a suitable reduced basis space with the many remaining "good" parameter samples, the main problem with such situations is the increased offline computational time for the additional SCM steps which try to achieve convergence also for the few "bad" samples.

Figure 4.8 shows the distribution of the parameter sample points chosen by the greedy algorithm for the generation of the reduced basis space and the maximum error in the energy norm over the parameter sample Ξ_{train} for the subsequent greedy steps. We note that the greedy sample points are not distributed equally but they are situated rather close to the boundaries of the parameter domain and thus follow the contour of the parameter domain. In the middle of the parameter domain, no sample points are chosen. As expected, the error reduction is not linear with the number of basis functions and the reduction rate decreases with an increasing number of basis functions. The reason for this is that for a higher number of basis functions, the solution can already be approximated relatively good without the new additional basis function and the error reduction is thus rather small compared to the error reduction in the first few greedy steps.

The presence of the ground close to the airfoil changes the behavior of the flow around the profile. This is demonstrated in figure 4.9. The left picture shows the asymmetry in the flow introduced by the ground effect. The velocities underneath the profile are increased and the rear stagnation point is moved slightly to the lower side of the profile while the front stagnation point remains on the nose of the airfoil. The maximum surface velocity is moved further to the tail on the lower side of the airfoil compared to the upper side. These observations are confirmed by the distribution of the pressure coefficient around the profile.

The relation of the real error between the FE and RB solution and the estimated error is illus-

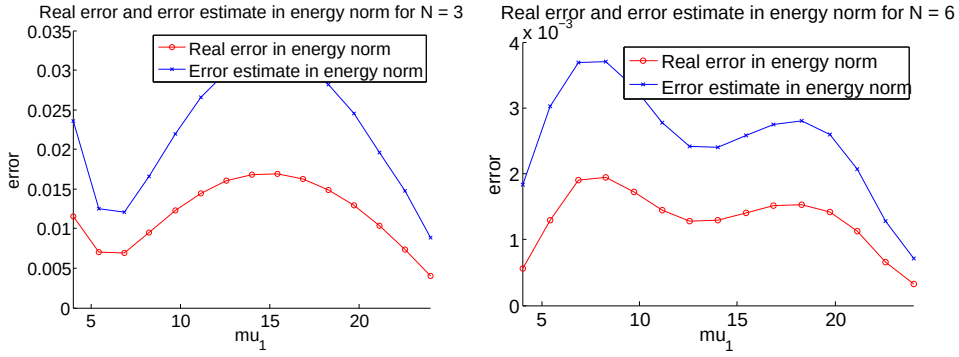


Figure 4.11: Real error and error estimate for N=3 (left) and N=6 (right) (SD).

	Big Domain	Small Domain
# elements mesh N_t	7013	6350
# subdomains	89	111
affine terms Q_a	30	34
affine terms Q_f	2	2
# RB functions $N_{\max}$	5	6
steps SCM $J_{\max}$	5	4
t_{FE}^{offline}	56min	2h 10min
t_{RB}^{offline}	1h 22min	2h 36min
$t_{FE}^{\text{online, 100}}$	9.00s	8.83s
$t_{RB}^{\text{online,100}}$	$1.02{\cdot}10^{-2}$s	$1.18{\cdot}10^{-2}$s
speedup	882	748

Table 4.6: Summary of the setups with zero angle of attack, with ground effect.

trated in figure 4.10 for this more complex variant. The sharpness and rigor of the error estimator are again verified and the variation of the error over the parameter domain is well represented by the error estimate. This property is crucial for the idea of the greedy algorithm. In figure 4.11, the development of the error for different numbers of RB basis functions is depicted. Here, μ_2 remains fixed while μ_1 varies over the whole parameter range D_1. It is obvious that not only the level of the error decreases as the number of basis functions increases, but also the locations of maximum and minimum errors change their positions as the solutions for parameter values close to the new greedy sample points are better approximated than others.

4.4.2 Problem 2: Single Profile, with Angle of Attack

Variant A: No Ground Effect

Also this problem is a two-parameters problem, but the second parameter is now the angle of attack. The parameter ranges and reference parameters are summarized in table 4.7. The specification of the boundary conditions depends on the particular realization. We considered three different implementations of the basic problem, as described in section 3.2. The first realization (a) implements the angle of attack in the boundary conditions. The implementation is then straightfor-

D_1	D_2	$\boldsymbol{\mu}_{\text{ref}}$	$\overline{\boldsymbol{\mu}}$
$[4, 24]$	$[0.01, \pi/4]$	$(14,\ \pi/8)$	$(14,\ \pi/8)$

Table 4.7: Parameter range and reference parameters for problem 2, variant A.

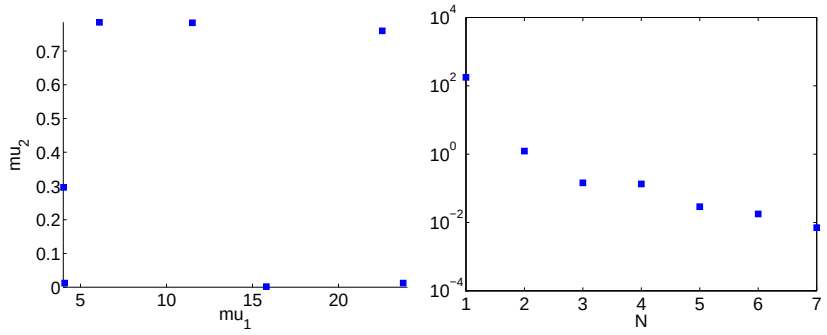


Figure 4.12: Sample points of the greedy RB space and maximum relative energy error as a function of the number of basis functions N for problem 2, variant A (a).

ward: We set $\frac{\partial \phi}{\partial n} = |\boldsymbol{u}_{\text{in}}| \cos(\mu_2)$ on the left domain boundary and $\frac{\partial \phi}{\partial n} = |\boldsymbol{u}_{\text{in}}| \sin(\mu_2)$ on the upper and lower domain boundary. As the prescription of Dirichlet boundary conditions dependent on the parameter and the spatial coordinates x_1 and x_2 was not yet possible with the version of the rbMIT software used for this chapter, we rotated the right domain boundary about the angle of attack and specified homogeneous Dirichlet conditions for the potential. Because the streamlines are always perpendicular to the equipotential lines, this has the same effect as the prescription of the analytically correct potential solution for the rotated velocity field on the original (not rotated) domain boundary. For the second realization (b), the boundary conditions are equivalent to problem 1, variant A, as the angle of attack is realized by a rotation of the airfoil. This is perhaps the most intuitive approach, since it reproduces the circumstances in real applications. The third realization (c) is similar to the first realization where the airfoil remains fixed. But now, the whole domain is rotated to follow equipotential respectively streamlines. This allows us to specify homogeneous Neumann conditions on the upper and lower domain boundary, inhomogeneous Neumann conditions on the left boundary and homogeneous Dirichlet conditions on the right boundary. Our aim is here to examine the influence of the implementation decisions on the reduced basis offline performance. The relevant results for all setups are given in table 4.8. For the resulting affine decompositions and meshes refer to appendix A.

First, we will again study the distribution of the greedy sample points in figure 4.12. Here, the distribution for the first setup (a) is depicted, but the distributions for the other two setups (b) and (c) show the same fundamental behavior, already observed for variant A: The sample points form a "contour region", that means that they are clustered close to the boundaries of the parameter domain.

On the basis of figure 4.13 (obtained with realization (a)), the influence of the second parameter μ_2, the angle of attack, on the solution shall briefly be discussed. We see that with an angle of attack different from zero, the flow field and pressure distribution is not symmetric anymore on the upper and lower surface of the airfoil, but gets nearly antisymmetric. The pressure peaks increase with increasing angle of attack. The limited physical nature of the potential model can also be observed: The streamlines go around the sharp trailing edge to form the rear stagnation point on the upper side of the profile. As mentioned before, in real flow situations the streamlines of the upper part of the geometry and those of the lower part would be parallel to the trailing edge angle and thus fulfill the Kutta-condition.

The comparison of the three setups in table 4.8 directly reveals the best implementation. The implementation of the rotated flow field in the boundary conditions requires not more effort than the previous problems, as the FE offline times t_{FE}^{offline} are comparable. The same can be stated for the RB offline time t_{RB}^{offline}. This setup is extremely advantageous, as most of the effect of the new parameter is "encapsulated" in the boundary conditions and the geometric variations remain relatively small. In contrast, the offline times both for the FE setup and the RB generation are

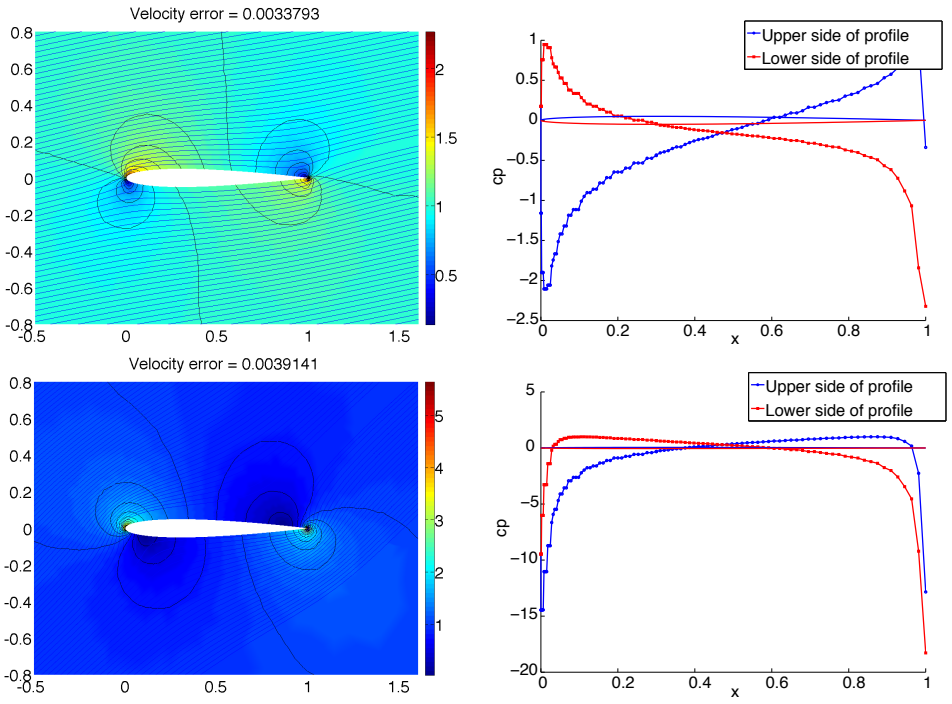


Figure 4.13: Streamlines with velocity magnitude and distribution of cp around the profile for a thickness of 10% and an angle of attack of 0.2 (upper) and 0.7 (lower).

extremely increased for the two other realizations, especially for the third setup (c). The reason for this is on the one hand the increased complexity of the geometric variations and, consequently, of the affine domain decomposition and the resulting affine representation (2.11). The higher offline times for the RB generation process can on the other hand be explained by the high number of affine terms Q_a (which is even higher than the number of subdomains in the affine decomposition) for setup (b) and by a combination of this problem and a non-converging SCM algorithm for setup (c). It is thus very important to keep in mind that there can be huge savings in the offline costs if the geometry specification is done in an advantageous way. However, the online performances do not differ much for the three different setups, since the number of basis functions is almost equal and the differences are only caused by the increased number of affine terms Q_a for the setups (b) and (c). Finally, we note that the non-convergence of the SCM algorithm for case (c) seems to

	Realization (a) Small Domain	Realization (b) Big Domain	Realization (c) Small Domain
# elements mesh N_t	5350	3313	3136
# subdomains	73	40	41
affine terms Q_a	22	45	38
affine terms Q_f	4	1	1
# RB functions $N_{\max}$	7	7	8
steps SCM $J_{\max}$	9	45	200
t_{FE}^{offline}	1h 9min	11h 24min	27h 31min
t_{RB}^{offline}	1h 39min	18h 27min	39h 18min
$t_{FE}^{\text{online, 100}}$	6.70s	5.20s	4.27s
$t_{RB}^{\text{online,100}}$	$8.78{\cdot}10^{-3}$s	$2.12{\cdot}10^{-2}$s	$1.32{\cdot}10^{-2}$s
speedup	764	245	325

Table 4.8: Summary of the setups with angle of attack included, no ground effect.

D_1	D_2	D_3	$\boldsymbol{\mu}_{\text{ref}}$	$\overline{\boldsymbol{\mu}}$
$[4, 24]$	$[0.01, 0.15]$	$[0.2, 2]$	(14, 0, 0.5)	(14, 0.075, 0.2)

Table 4.9: Parameter range and reference parameter for problem 2, variant B.

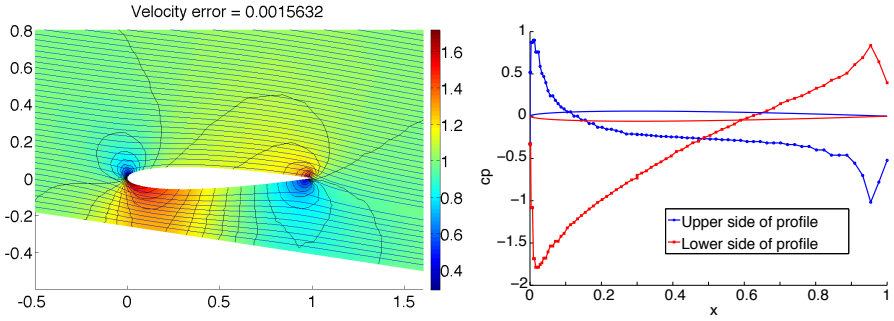


Figure 4.14: Streamlines with velocity magnitude and distribution of cp around the profile for a thickness of 12%, an angle of attack of 0.15 and a distance from ground of 0.2

be of the uncritical type described before, where a lower bound for the coercivity constant can be obtained with sufficient accuracy for almost all parameter samples in the set $\Xi_{\text{train,SCM}}$ and only one or a few parameter samples lead to divergence. This conclusion can be drawn of the fact that the number of basis functions is not increased dramatically compared to the other setups, that means that the remaining "good" setups are sufficient for the greedy algorithm to work satisfactory.

Variant B: Ground Effect included

We now consider the combination of problem 1, variant B with problem 2, variant A. The resulting problem has three geometric parameters, the thickness parameter μ_1, the angle of attack μ_2 and the distance from the ground μ_3. μ_3 is now defined differently to problem 1, variant B as the distance between the lowest point of the profile with zero angle of attack to the ground ($x_2^l = -\frac{\mu_1}{200} - \mu_3$). Again, two different implementations of this problem equivalent to the setups (a) and (b) of variant A are realized. The major changes compared to setup (a) of variant A is that now also the lower domain boundary has to be rotated in order to coincide with the ground and homogeneous Neumann conditions have to be prescribed on this boundary. Setup (b) differs from setup (b) in variant A only by the fact that the ground is moved towards the profile according to the distance parameter (here, the distance is again defined as for problem 1, variant B). This geometry, in both realizations, is much more demanding for the software, especially in the symbolic preprocessing steps. On the one hand, the affine decomposition is very demanding regarding memory aspects due to the complex geometric variations introduced by the interaction of the three different parameters, and on the other hand the risk of subdomains getting singular for parts of the parameter domain is increased considerably. It was therefore especially important to reduce the parameter ranges as much as possible and to choose the reference parameters very carefully. The successful choices are summarized in table 4.9.

The main topic of our examinations for this problem is again the comparison between the real error and the error estimate. In addition, more insight into the affine transformations of the geometries and the associated FE mesh shall be given. Before we concentrate on these topics, an overview of the reduced basis offline generation process shall be given. The affine domain decompositions and resulting FE meshes for both setups are again shown in appendix A. To give an impression of the flow phenomena for this more complicated geometric setups where several different effects overlap and influence each other, the flow field and the distribution of c_p around the profile are illustrated in figure 4.14 for $\boldsymbol{\mu} = (12, 0.15, 0.2)$. This is an extremal case, as the

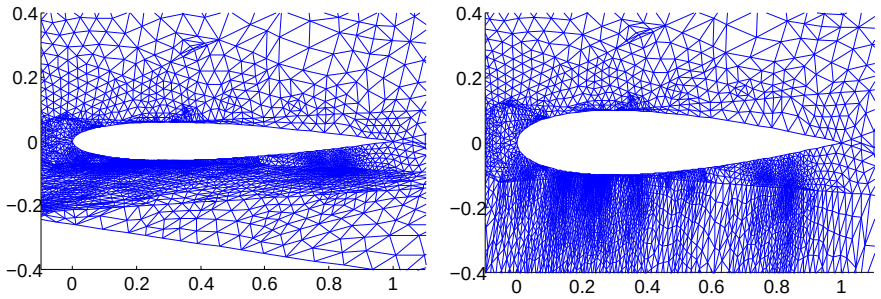


Figure 4.15: Details of the meshes for $\boldsymbol{\mu}([12, 0.15, 0.2)$ (left) and $\boldsymbol{\mu} = (20, 0.1, 2)$ (right).

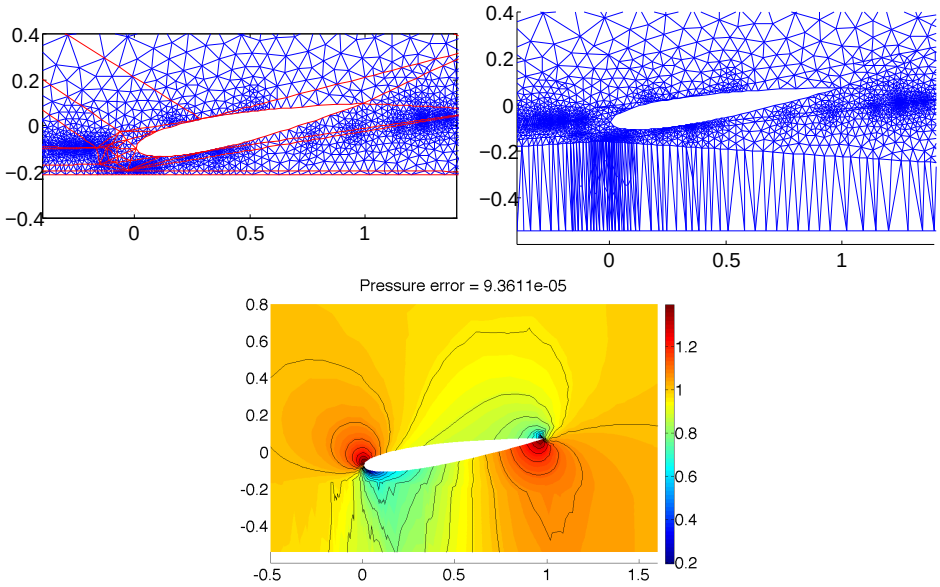


Figure 4.16: Low quality FE mesh for the reference parameter $\boldsymbol{\mu}_{\text{ref}}$ (upper left) and for $\boldsymbol{\mu} = (12, 0.15, 9)$ (upper right); pressure distribution for $\boldsymbol{\mu} = (12, 0.15, 9)$ (lower).

maximum angle of attack is combined with the minimum distance. The velocity is increased between the profile and the ground due to the presence of the ground and the rotation even increases this effect. The surface pressure still has an antisymmetric character, but the suction peak on the lower side of the profile is much higher than the opposed suction peak on the upper side of the profile. The rear stagnation point is moved in the direction of the nose on the lower side of the airfoil, compared to the case where ground effects are not important.

We will now discuss the effect of the parameter domain and the reference parameters on the quality of the (deformed) meshes and thus on the underlying FE solution. In figure 4.15, the deformed meshes for the parameter instances $\boldsymbol{\mu} = (12, 0.15, 0.2)$ and $\boldsymbol{\mu} = (20, 0.1, 2)$ are depicted. As the mesh is very fine in the critical region between the profile and the ground, where the deformations and distortions are stronger than in the other regions of the domain, the quality of the deformed meshes is still sufficient and we can expect a relative accurate FE solution for these meshes. This in turn ensures that the results of the reduced basis computations approximate well not only the FE solution, but also the solution of the exact problem, which is the ultimate goal and should not be forgot. The mesh presented in figure 4.16 in turn is an example for a miserable choice of the parameter domain and the reference parameter. While the problems are not visible in the reference configuration, the picture of the deformed mesh reveals that the number of elements between the profile and the domain boundary are not sufficient. The affine transformation leads to

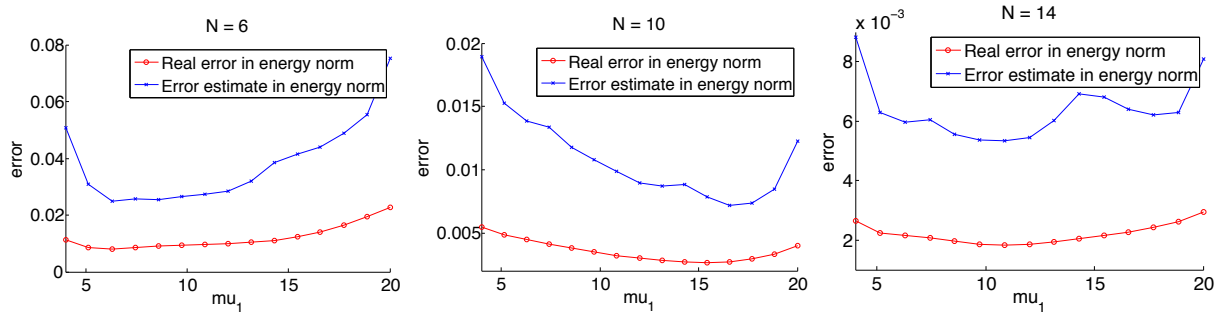


Figure 4.17: Real error and error estimate for $N = 6$, $N = 10$ and $N = N_{\max} = 14$.

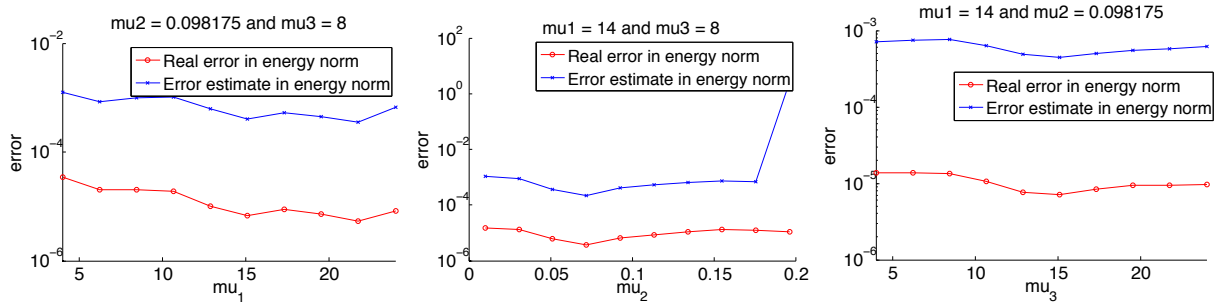


Figure 4.18: Real error and error estimate for varying μ_1, μ_2 and μ_3.

a strong stretching in this region. The problem is that in the reference configuration this region is very narrow and is therefore only filled with a few FE mesh elements which are extremely stretched and therefore reduce the accuracy of the FE solution dramatically, as can be seen in figure 4.16 (c). To avoid such effects, the reference parameter for the distance to the ground has been chosen relatively high and the parameter range for the distance to the ground has been adjusted.

The second emphasis in this section is the comparison of the real error with the RB error estimate. In figure 4.17 we first concentrate on the reduction of the error with an increasing number of basis functions. The distance to the ground and the angle of attack remain fixed, while the thickness varies over the whole parameter range D_1. We note that the error is decreasing over the whole domain with increasing number of basis functions, while the location of the maximum errors changes according to the choices of the sample points by the greedy algorithm. It is also interesting to observe that the true error is a lot smoother than the estimated error in this example, but the position of the maximum error is predicted rather reliably. As a general remark, the error estimate is extremely sharp. As we shall see in the subsequent sections, the error bounds and the real error for other fluid models differ very often by several orders of magnitude.

The last pictures illustrate the very interesting case of realization (b), where the SCM algorithm does not converge in the prescribed maximum number of steps. Again, the true error between FE and RB solution is compared with the RB error estimate. We let vary μ_1, μ_2 and μ_3 in turn while the other two parameters remain fixed. The error estimate has been proven to be very sharp if the SCM algorithm is able to compute sharp lower bounds α_{LB} for the coercivity constant. In this case, where the lower bounds for the coercivity constant have not converged yet, the estimated error differs by several orders of magnitude from the true error which results in an unnecessarily augmented number of basis functions for the reduced basis space, as we can observe in the comparison of setup (a) and (b) in table 4.10. In the middle figure for varying μ_2 and fixed μ_1 and μ_3, a possible problematic value can be found: For $\mu_2 \approx 0.2$, the estimated error is a lot higher than for the rest of the parameter domain. The reason for this is that the SCM algorithm can not calculate α_{LB} for this parameter sample due to a bad numerical situation for exactly this value. For the remaining

	Realization (a) Small Domain	Realization (b) Big Domain
# elements mesh N_t	6260	4271
# subdomains	57	40
affine terms Q_a	37	42
affine terms Q_f	2	2
# RB functions $N_{\max}$	14	54
steps SCM $J_{\max}$	55	200
t_{FE}^{offline}	1h 17min	38h 29min
t_{RB}^{offline}	4h 37min	65h 46min
$t_{FE}^{\text{online, 100}}$	9.31s	6.56s
$t_{RB}^{\text{online,100}}$	$1.73{\cdot}10^{-2}$s	$8.03{\cdot}10^{-2}$s
speedup	538	82

Table 4.10: Summary of variants with angle of attack and ground effect included.

D_1	D_2	D_3	D_4	$\boldsymbol{\mu}_{\text{ref}}$	$\overline{\boldsymbol{\mu}}$
$[4, 24]$	$[0.01, \pi/16]$	$[0.85, 1]$	$[1.2, 1.6]$	(14, $\pi/16$, 0.85, 1.2)	(14, $\pi/16$, 1, 2)

Table 4.11: Parameter range and reference parameters for problem 3.

parameter domain, the coercivity constant might be approximated already very well, while for this parameter (and probably also several other discrete parameter values), the lower bound remains far from the true coercivity constant. The main difference between the (unproblematic) case of variant A, setup (c) and the (problematic) case for the current setup is that the lower bounds for the problematic values are either negative or positive but not close to zero in the first case, while they are positive and very close to zero in the latter case. The greedy algorithm is then unable to recognize the problematic values as "bad" sample points and will try to reduce the (overestimated) error for these parameter instances although the real error is not higher than for the surrounding parameter values. In the unproblematic case, the "bad" sample points can either be excluded from the greedy sampling process or the influence of the overestimated errors is relatively small and does not lead to an (extremely) increased number of basis functions.

4.4.3 Problem 3: Two Profiles

In the third problem, we consider a geometry which involves two profiles of equal thickness with four geometrical parameters. The first parameter is, as usual, the thickness of the profiles μ_1. The second parameter is the angle of attack μ_2. The third and fourth parameters μ_3 and μ_4 are

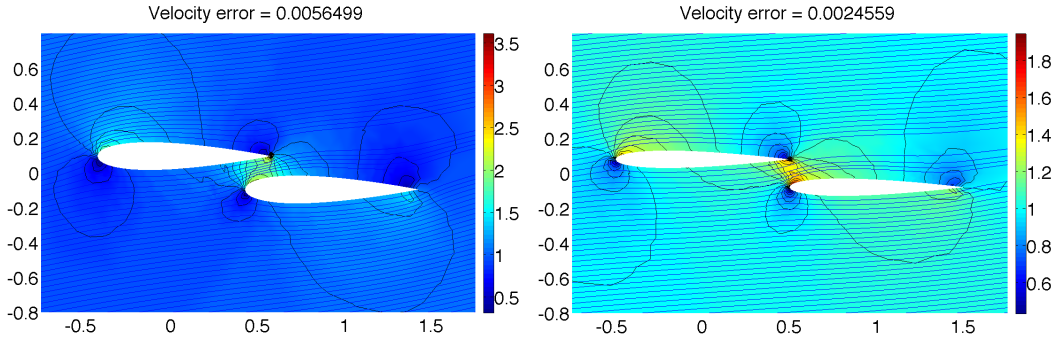


Figure 4.19: Pressure distribution and streamlines/velocity magnitude for $\boldsymbol{\mu} = (16, \pi/16, 0.85, 1.0)$ and $\boldsymbol{\mu} = (10, \pi/32, 1.0, 1.6)$.

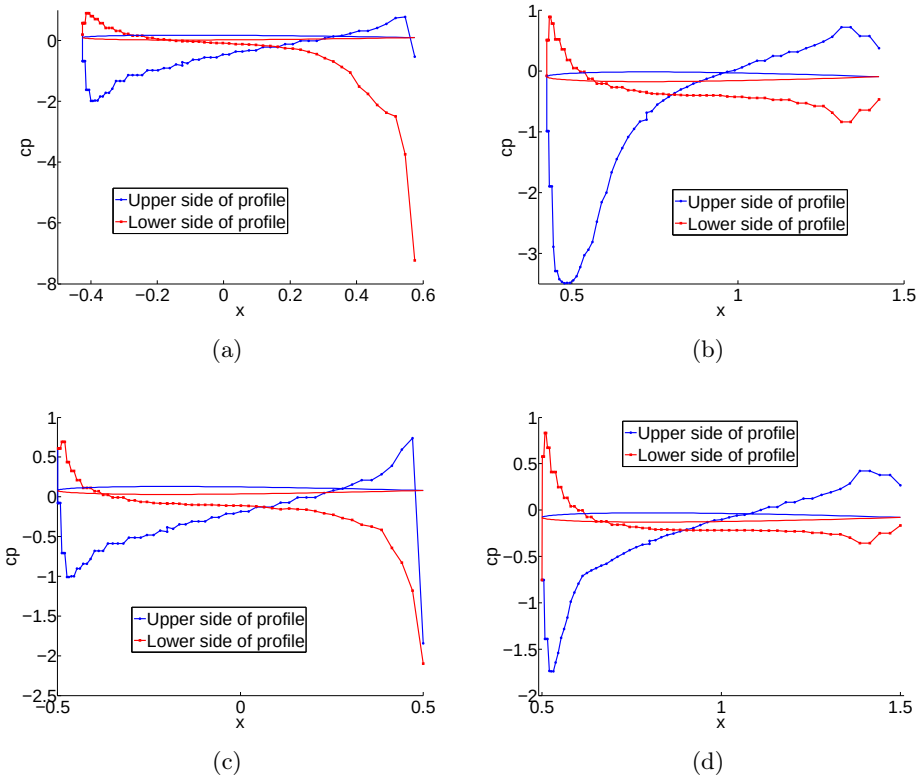


Figure 4.20: Distribution of c_p around the profiles for $\boldsymbol{\mu} = (16, \pi/16, 0.85, 1.0)$ (left profile (a) and right profile (b)) and for $\boldsymbol{\mu} = (10, \pi/32, 1.0, 1.6)$ (left profile (c) and right profile (d)).

the distance of the profiles in x_1- and x_2-direction, respectively. The distance in x_2-direction is specified in terms of half of the thickness to make it easier to ensure a certain minimum distance of the profiles without restricting the parameter range too much. This problem has been very challenging in regard to the correct specification of parameter range and reference parameters because the geometric variations due to the many parameters involved are very complex. The profiles have to be situated relatively close to each other to make them interact. This in turn makes the domain decomposition in the region between the profiles very complicated and the risk of subdomains becoming singular or overlapping even for parameters very close to the reference parameters is huge. Also the deformations in the mesh in that region are relatively strong. The chosen parameter ranges and reference parameters are indicated in table 4.11. The boundary conditions are identical to problem 2, variant A.

For the affine decomposition of the whole domain and the FE mesh, we refer again to appendix A. Note from table 4.12 that the affine domain decomposition in this case results in 163 subdomains, which can only be reduced to $Q_a = 102$ affine terms in the bilinear form. These numbers are very high and lead to a less efficient computational procedure both in the offline and online stage. However, taking into account that this parametrized geometry is very complex and involves not only many parameters and two profiles (described by curved lines of 8th order!), but also very small and narrow regions between the two airfoils, this result is already remarkably good. Note that for this problem, the greedy algorithm needs only a relatively moderate number of 20 basis functions, although the SCM process does not converge completely.

The effect of the two profiles interacting with each other on the flow field is illustrated in figure 4.19 for two different parameter values. The changes of the surface pressure of the two airfoils due to the interaction can be better demonstrated by means of the distributions of the pressure

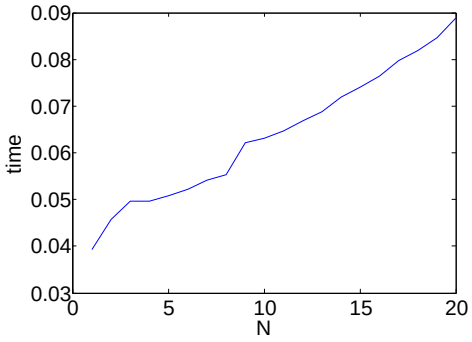


Figure 4.21: Dependency of the RB online time on the number of basis functions used for the solution of the problem.

coefficient c_p around the profiles, shown in figure 4.20. The suction peak on the lower side of the tail region of the front profile gets much stronger, while the suction peak on the upper side in the nose region of the rear profile increases as the velocities between the airfoils are higher compared to the flow around a single profile. The antisymmetric behavior of a single profile with an angle of attack different from zero can not be observed anymore. The flow pattern on the upper side of the front profile and on the lower side of the rear profile are not influenced much.

We will conclude this chapter with a short discussion of the computational times involved in this setup. Table 4.12 indicates a FE offline time of t_{FE}^{offline} =7h 50min. Compared to the number of subdomains that have been created, this is relatively quick. However, the RB offline time t_{RB}^{offline} =34h 34min is very long due to the non-converging SCM procedure and the high number of affine terms which makes every step in the SCM and greedy procedures more costly. Nevertheless, the reduced basis method is still almost 200 times faster in the online computations as the FE method. This ratio could be improved further especially if the number of affine terms could be reduced. Figure 4.21 illustrates the dependence of the RB online time $t_{RB}^{online,100}$ of the number of RB basis functions used to solve the problem. It has to be mentioned here that these computations have been made on another computer than the comparisons between RB and FE online performance which explains the difference in absolute computational time. It becomes clear that the dependence on the number of basis functions is linear, which is in good agreement with the theory.

	Problem 3
# elements mesh N_t	3693
# subdomains	163
affine terms Q_a	102
affine terms Q_f	3
# RB functions $N_{\max}$	20
steps SCM $J_{\max}$	200
t_{FE}^{offline}	7h 50min
t_{RB}^{offline}	34h 34min
$t_{FE}^{\text{online, 100}}$	17.15s
$t_{RB}^{\text{online,100}}$	$8.80 \cdot 10^{-2}$s
speedup	195

Table 4.12: Summary of problem 3: two profiles.

5 The Reduced Basis Method for Thermal Flows around an Airfoil

In the previous chapter, we applied the reduced basis method to potential flows. We concentrated thereby on the affine geometry decomposition and the reduced basis solution and a posteriori error estimation for the field solution of symmetric problems. These aspects shall now be extended by the output approximation and a posteriori output error estimation considering noncompliant and nonsymmetric problems within the reduced basis method. We will therefore apply the reduced basis method to thermal flows governed by a convection-diffusion equation on some of the parametrized geometries described in section 3.2. The necessary fundamentals and procedures for these extensions have been established in sections 2.6.1 and 2.6.2.

In the following sections, we will first introduce the concept of thermal flows described by convection-diffusion equations. The next step will be again the reformulation of the necessary equations in order to obtain a problem which is equivalent to (2.1). Finally, the numerical results for the different geometrical setups will be discussed.

5.1 The Convection-Diffusion Equation

We are considering a class of thermal flows where the transport of scalar variables such as the temperature is induced by diffusive effects in combination with convection effects due to a present flow field, but the temperature distribution itself has no direct influence on the flow field (see section 3.3). That means that, as there is no direct feedback of the thermal conditions on the flow field, we can first compute the flow field with a suitable (incompressible) fluid model and subsequently compute the temperature distribution (or the distribution of other scalar variables, such as concentrations) based on the flow field. Such thermal flows are governed by the convection-diffusion equation (see e.g. [3, 4, 62, 70]):

$$-\frac{1}{\mathrm{Pe}}\Delta T + \mathbf{u}\cdot\nabla T = 0 \qquad \text{in } \Omega_o(\boldsymbol{\mu}) \tag{5.1}$$

T is the transported scalar (in the following, we will always interpret T as a temperature), while $\mathbf{u}$ is the velocity of the free stream that "convects" the scalar. Here, Pe is the Peclet number. It indicates the relation of the diffusive term ΔT to the convective term $\mathbf{u}\cdot\nabla T$. If we assume that $|\mathbf{u}| = 1$ and $\mathrm{Pe} \ll 1$, the diffusive term is more important. If $\mathrm{Pe} \gg 1$, the convective transport is dominating.

A special property of flows governed by convection-diffusion equations is that the character of the solution changes dramatically with Pe. Especially for high Peclet numbers, very thin boundary layers can occur. This leads to the known stability problems when approximating a convection diffusion problem with the FE method for higher Peclet numbers. Then either the mesh has to be refined to resolve these boundary layers, or special stabilization techniques have to be applied. The reduced basis methodology itself does not make such demands (as smoothness in the spatial domain is no necessary requirement), but it is fully transparent to the underlying FE solution. This means that if the underlying FE solution technique, on which the RB process is based, is stable, the RB solution will remain stable. If the FE solution is unstable and shows oscillations, the RB solution will also do. If the FE solution is only oscillating slightly for a part of the parameter

domain, this need not necessarily spoil the RB solution for the whole parameter domain, but it will certainly increase the number of RB basis functions needed to reproduce the FE solutions properly. Therefore it has to be made sure that the underlying FE method is chosen properly or that the parameter domain is bounded in a way that allows the FE solution to remain stable. More information about the FE treatment of convection-diffusion problems can be found e.g. in [14] and in [15].

The convection-diffusion equation is not a "classical" model applied in the domain of fluid flow around wing profiles for F1 race cars. Nevertheless, the correct computation of the convection-diffusion scalar case is critical for the application of the Navier-Stokes equations and therefore an interesting intermediate step (where the convective field is given or computed by the solution of a transport problem), as the dualism between convective and diffusive effects is also relevant for Navier-Stokes equations. For the significance of the convection-diffusion equations in the reduced basis context see also section 3.3.

In the RB domain, a convection-diffusion problem in a similar context involving parametrized geometries with rotating blades (but for the nonaffine case) has been studied by Tonn and Urban in [74]. Recently, Dedè considered the application of the RB method to the related advection-reaction problems in [13]. More examples for the treatment of convection-diffusion problems with the RB method can also be found e.g. in [61] and [68].

The convection-diffusion equation is, as the Poisson equation for the potential flows, a scalar equation that allows us to remain in the affine context. But in this case, the resulting bilinear form of the weak formulation is no longer symmetric. This means that our problem is no longer compliant. Therefore, the extensions developed in section 2.6 are needed for the efficient solution of convection-diffusion problems with the reduced basis method.

5.2 Problem Statement

The weak form of (5.1) can be derived by the same procedure as applied for the Laplace equation in section 4.2. We first recall the function space X^e and associated norms of section 2.2.1 and multiply (5.1) by a test function $v \in X^e$. The resulting expression is then integrated over $\Omega_o(\boldsymbol{\mu})$. If we apply integration by parts and the divergence theorem, the variational formulation of (5.1) follows as

$$\frac{1}{\mathrm{Pe}} \int_{\Omega_o(\boldsymbol{\mu})} \nabla T \cdot \nabla v + \int_{\Omega_o(\boldsymbol{\mu})} (\boldsymbol{u} \cdot \nabla T) v - \int_{\partial\Omega_o^N(\boldsymbol{\mu})} \frac{\partial T}{\partial n} v \qquad \forall v \in X^e, \tag{5.2}$$

where $\partial\Omega_o^N(\boldsymbol{\mu})$ is the part of the boundary on which inhomogeneous Neumann conditions

$$\frac{\partial T}{\partial N} = g_N \tag{5.3}$$

are prescribed.

From this, we obtain the weak statement as

$$a(w, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in X^e, \tag{5.4}$$

with

$$a(w, v; \boldsymbol{\mu}) = \frac{1}{\mathrm{Pe}} \int_{\Omega_o(\boldsymbol{\mu})} \nabla w \cdot \nabla v + \int_{\Omega_o(\boldsymbol{\mu})} (\boldsymbol{u} \cdot \nabla w) v \tag{5.5}$$

$$f(v; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} \frac{\partial w}{\partial n} v. \tag{5.6}$$

The formulation on the reference domain Ω (2.1) can again be obtained from the formulation

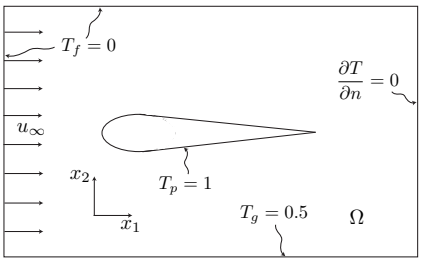


Figure 5.1: Bondary conditions.

(5.4) on the parameter-dependent domain $\Omega_o(\boldsymbol{\mu})$ with the methods of section 2.7.

5.3 Numerical Results

In this section, we will present the results obtained by applying the reduced basis method as implemented in the rbMIT software package to the convection-diffusion equation introduced above. Again, the standard settings introduced in table 3.1 are used. We consider the thermal flow in the parametrized geometries of section 3.2. All problems are now based on the rectangular "big domain" of the size $[-6, 6] \times [-4, 4]$. In the setups with ground effect, the lower domain boundary is moved closer to the profile. The boundary conditions for the temperature for all variants are chosen similarly, see figure 5.1. The nondimensional temperature of the surface of the profile is always set to $T_p = 1$ (inhomogeneous Dirichlet conditions), while the temperature T_f of the free stream on the boundaries (except of the right domain boundary) is set to zero (homogeneous Dirichlet conditions). In the variants involving ground effects, the lower domain boundary, representing solid ground, is set to a temperature of $T_g = 0.5$ (inhomogeneous Dirichlet conditions). On the right domain boundary, we applied natural (homogeneous Neumann) outflow boundary conditions. This setup can for example represent a hot profile moved through cold air over a warm surface which has a temperature between the temperature of the profile and the temperature of the air. The value and direction of the free stream velocity is changing for the different setups and will therefore be mentioned for each setup below.

We are now concentrating on the capabilities of the RB method to compute a scalar output based on the field solution in a rapid and reliable way. As output function we consider the integral of the temperature over the right (outflow) domain boundary. By dividing this value by the length of this boundary, we can easily compute the mean temperature on the outflow boundary. This output is thus not too abstract and related to the field solution in a relatively simple way.

5.3.1 Problem 1: Single Profile, no Angle of Attack

Variant A: No Ground Effect

In this variant, the free stream approaches the profile with zero angle of attack: $\mathbf{w} = (1, 0)^T$. The parameters for this problem are the thickness of the profile μ_1 and the Peclet number μ_2. The thickness may vary in $\mu_1 \in [4, 24]$ while the Peclet number has to be in $\mu_2 \in [0.1, 100]$. The range for the second parameter is very large, since the behavior of the solution changes completely for the extremal values of $\mu_2 = 0.1$ and $\mu_2 = 100$.

The affine decompositions and FE meshes obtained for this and the following problems are very similar to the affine decompositions and FE meshes obtained for the potential flows discussed in the previous chapter. We will therefore not discuss the details here. For the relevant parameters, such as the the number of affine terms Q_a and the number of mesh elements N_t, refer to table 5.1.

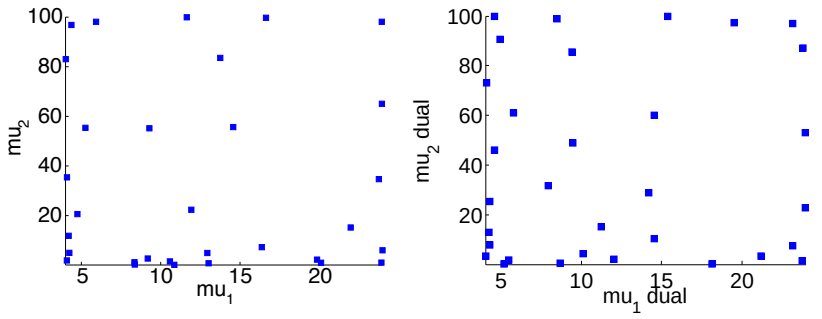


Figure 5.2: Sample points of the primal (left) and dual (right) greedy RB space.

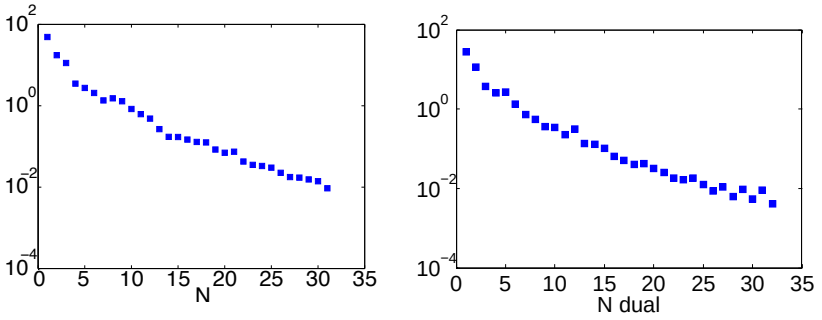


Figure 5.3: Maximum relative energy error as a function of the number of basis functions N for the primal (left) and the dual (right) problem.

We note that the linear form has zero terms for all four problems that will be presented in this chapter. The decompositions and meshes are shown in appendix A.

As this problem is nonsymmetric, we do not compute only the primal RB approximation space but also a dual RB approximation space is necessary. The greedy algorithm selects $N_{pr} = 31$ basis functions for the primal and $N_{du} = 32$ basis functions for the dual approximation space. The sample point distributions for the primal and the dual problem are visualized in figure 5.2. It is interesting to note that the distributions are very similar and both show (as for the potential flows) a higher density close to the boundaries of the parameter domain. Especially in the primal procedure, the highest density is obtained for small values of μ_2. As we will see below, this is related to the higher sensitivity of the problem to this parameter for small values than for high values: The changes in the solution are stronger for small changes of μ_2 around $\mu_2 = 0.1$ than around $\mu_2 = 100$. Therefore, the resulting sample point density has a logarithmic character in μ_2.

The corresponding maximum relative energy errors (for the field solution) for the primal and the dual problem are shown in figure 5.3. The error tolerance for the dual problem is half of the error tolerance for the primal problem. The error reduction rate is very similar, but the dual error decreases slightly faster and starts to show some oscillations for higher N. Nevertheless, the resulting number of basis functions is comparable, as it is the case for all four problems investigated.

To illustrate the influence of the Peclet number on the solution of the problem, the temperature fields for four different parameter combinations are shown in figure 5.4. It can be observed that the diffusion indeed dominates for $\mathrm{Pe} = 0.1$. The influence of the convecting free stream is barely noticeable and the influence of the hot profile stretches very far, both up- and downstream as well as across the stream. With increasing Peclet number, the convection of the temperature distribution with the free stream to the right gets stronger and the temperature boundary layers get thinner until the presence of the profile is only noticeable downstream in a very small corridor of little more than the thickness of the profile.

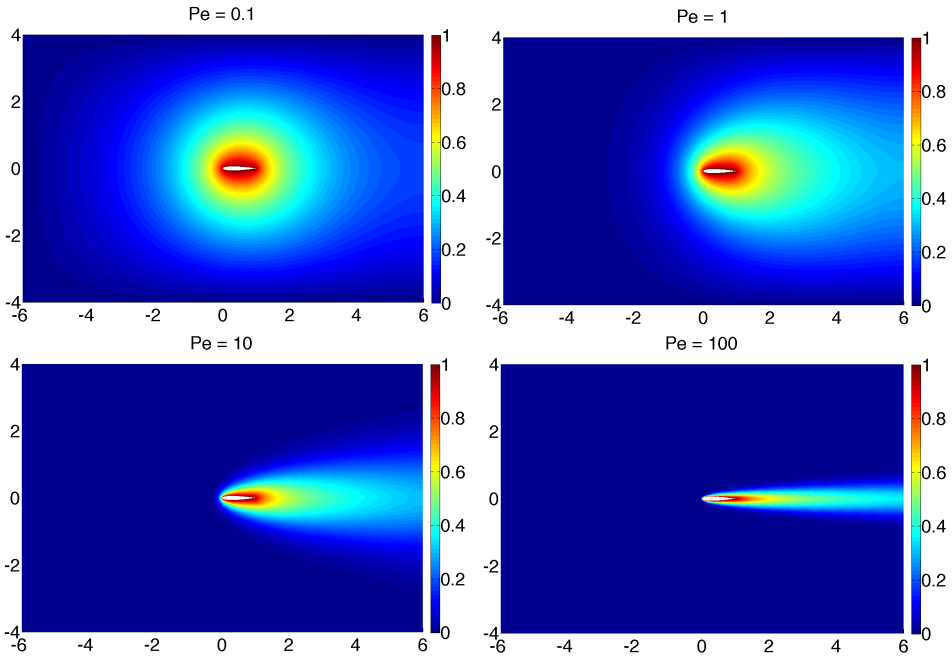


Figure 5.4: Temperature distribution for a thickness of $\mu_1 = 12$ and Peclet numbers of Pe = 0.1, Pe = 1, Pe = 10 and Pe = 100 (from upper left to lower right).

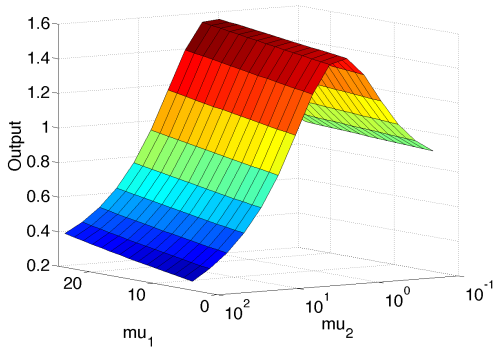


Figure 5.5: Variation of the output functional over the parameter domain.

We will now concentrate on the output computation. In figure 5.5, the variation of the output (the integrated temperature on the outflow boundary) for the whole parameter domain is shown. The logarithmic scaling of the μ_2 axis shows that the variations are indeed very strong for small Peclet numbers while they are less pronounced for the bigger Peclet numbers. It becomes also clear that the influence of the thickness is relatively small compared to the influence of the Peclet number.

Figure 5.6 compares the a posteriori error estimates for the output with the real error between RB and FE output on the whole parameter domain. It becomes clear that the highest errors occur for small values of the Peclet number, which is reflected well by the error estimate. However, the error estimate is not very sharp for this problem, although the SCM algorithm converged in a few steps to fulfill the requirements on the maximum ratio $\alpha_{tol} = \frac{\alpha_{UB}-\alpha_{LB}}{\alpha_{UB}} = 0.85$. We expected this behavior, because in the general non-compliant case, we can no longer obtain an upper bound for the output error bound effectivity. Hence we anticipate larger effectivities and thus more erratic error bounds than in the compliant case [68]. In this case, the real error and the error estimate may differ by about two to three orders of magnitude.

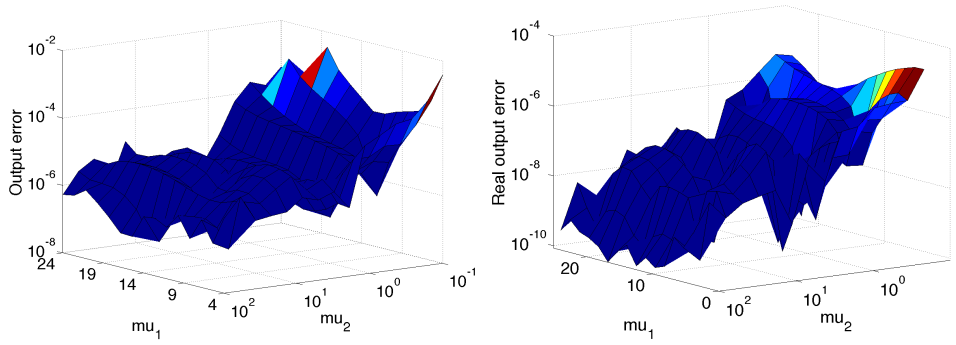


Figure 5.6: A posteriori output error estimate (left) and actual output error (right)

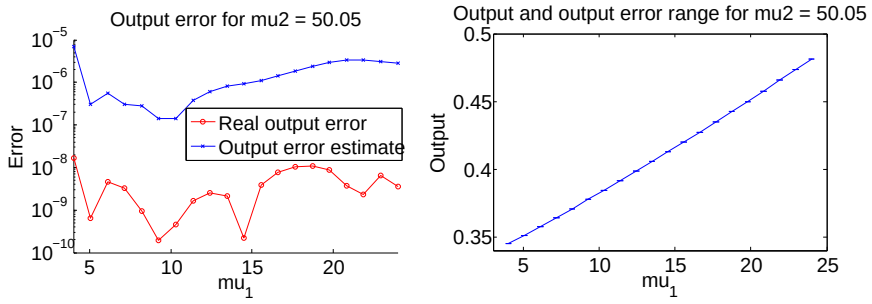


Figure 5.7: Output error and output with error bar interval (based on a posteriori output error estimate) for $\mu_2 = 50.05$, variable μ_1, for $N_{pr} = N_{\max}$ and $N_{du} = N_{\max}$.

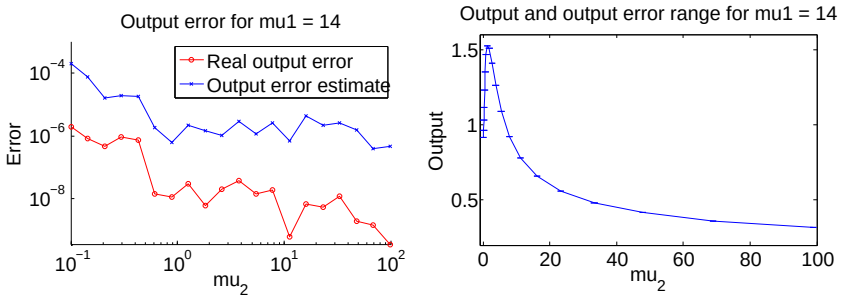


Figure 5.8: Output error and output with error bar interval (based on a posteriori output error estimate) for $\mu_1 = 14$, variable μ_2, for $N_{pr} = N_{\max}$ and $N_{du} = N_{\max}$.

For an easier evaluation of the output errors, figure 5.7 and figure 5.8 show the comparisons of the output error estimates with the real output error and the output distribution together with the error bar intervals for μ_2 respectively μ_1 fixed. Here, the maximum number of basis functions has been used both for the primal and for the dual problem. The variation of the output with the thickness of the profile is more or less linear, while for increasing Peclet number, it increases at first until Pe ≈ 2 and then decreases again. As mentioned before, the variation with the Peclet number is a lot stronger than the variation with the thickness. The errors and error estimates are very small, but the difference between the estimated and the real error varies between two and four orders of magnitude. This means, that the number of basis functions could be considerably reduced to keep the desired accuracy. The highest errors (both estimated and real) are associated with the small Peclet number limit, as the strongest changes in the output occur in this range[1].

[1]Note that the error between the FE solution and the exact solution behaves the other way round, since the

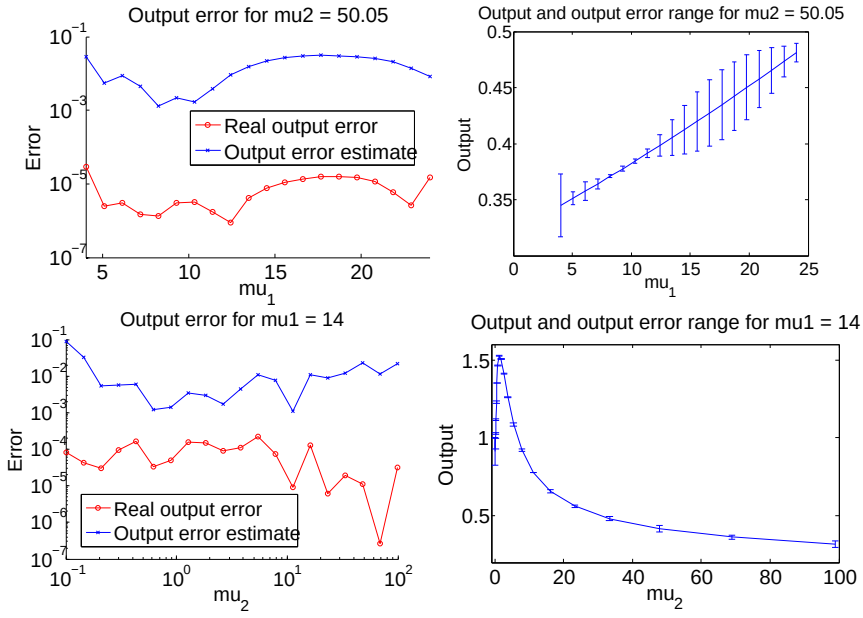


Figure 5.9: Output error and output with error bar interval (based on a posteriori output error estimate) for $N_{pr} = 0$ and $N_{du} = 30$.

In the following, the influence of the number of basis functions used to solve the primal and dual problem shall be investigated. For this reason, the total number (primal+dual) of basis functions is reduced to be $N_{tot} = 30$ and different combinations of N_{pr} and N_{du} are used to solve the problem. For each combination, the comparison of the error estimate with the real error and the output together with the the error bar intervals are shown. In figure 5.9 and in figure 5.10, the two extremal cases are shown, where only the primal or only the dual solution is used. Figure 5.11 shows the case when the number of primal and dual bases is equal. We note that the error estimates change considerably for the different combinations. It is obvious that the combination of the primal and dual RB approximation spaces clearly improves the accuracy of the RB approximation compared to the solution based on either the primal or the dual approximation space. These results are not surprising and state our theoretical assumptions of section 2.6.1 where the primal-dual framework has been developed explicitly to reach this aim.

As we are only comparing the solutions on two "lines" of the parameter domain, this result is of course not necessarily as pronounced for the entire parameter domain. In practical applications, a lookup table could be prepared in the offline phase, based on the greedy sample set which contains the maximum output error as a function of the number of primal and dual basis functions. This table could be used to find the optimal primal/dual number of basis functions pair to reach a desired accuracy, as mentioned in [68].

Finally, the offline and online times should be mentioned here to give an impression of the performance of the RB method for this case. The additional measures for the computational times introduced in this chapter are summarized in table 5.2. For this geometry setup, the FE offline time has been $t_{FE}^{\text{offline}} = 9min$, which is very short. The primal offline time for the RB method contains the FE offline time and the time necessary to generate the primal RB function space (the time to perform the SCM algorithm to calculate the lower bounds for the coercivity constant and the time to calculate the suitable primal basis functions by the greedy algorithm). This has been $t_{RB,pr}^{\text{offline}} = 44min$ which is still very short, considering the very fine mesh of the problem. The dual

numerical difficulties (concerning stability) in this case arise for high Peclet numbers while small Peclet numbers are uncritical and can be well approximated.

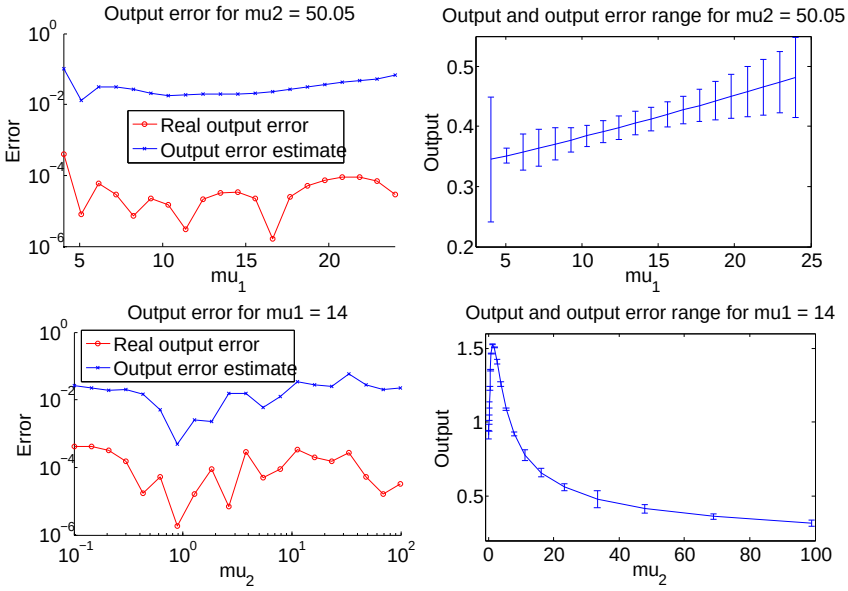


Figure 5.10: Output error and output with error bar interval (based on a posteriori output error estimate) for $N_{pr} = 30$ and $N_{du} = 0$.

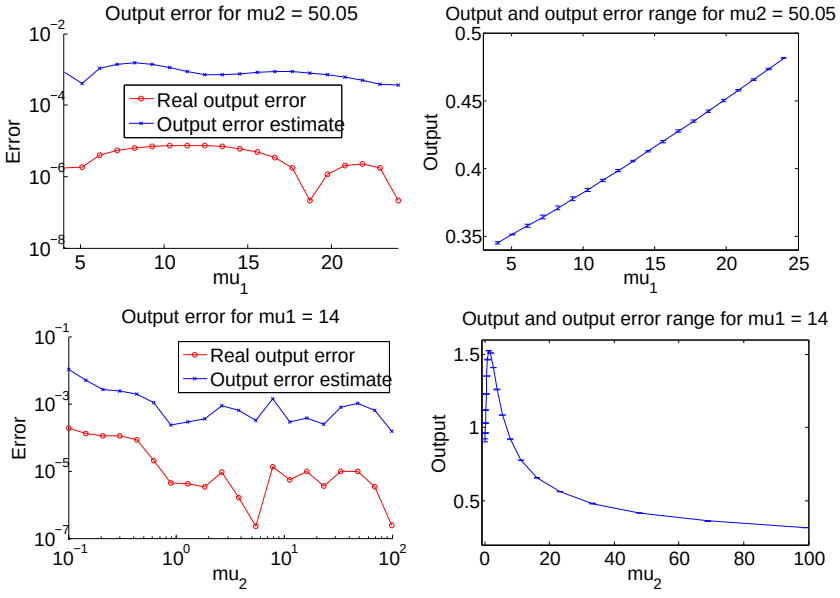


Figure 5.11: Output error and output with error bar interval (based on a posteriori output error estimate) for $N_{pr} = 15$ and $N_{du} = 15$.

offline time for the RB method contains the primal offline RB time plus the time to generate the dual RB space. In this case, it has been $t_{RB,du}^{\text{offline}} = 57min$ and thus the additional effort for the generation of the dual approximation space is definitely affordable. The FE online time for the computation of 100 temperature field solutions is $t_{FE}^{\text{online, }100} = 38.528s$, while the corresponding time with the RB method is $t_{RB}^{\text{online, }100} = 4.36 \cdot 10^{-2}s$ and thus it is almost 900 times faster. This performance is due to the very fine mesh and the moderate number of basis functions and affine terms in the bilinear form. In addition, we measured the time to compute the scalar output for 100 parameter instances with the full primal and dual RB space. In this case, this is $t_{RB,out}^{\text{online, }100} = 0.1544s$. Of course, the accuracy of these solutions is much higher than required and the performance should be improved by a lookup table as mentioned before.

Variant B: Ground Effect Included

The free stream velocity in this setup is as in variant A horizontal. The fact, that the velocity should decrease to zero on the ground has not been taken into account here and thus it is again set to $\mathbf{w} = (1, 0)^T$. The parameters for this problem are the thickness of the profile $\mu_1 \in [4, 24]$, the relative distance to the ground $\mu_2 \in [1.5, 3]$ in terms of the thickness and the Peclet number $\mu_3 \in [0.1, 100]$.

We note from table 5.1 that the SCM algorithm is less efficient for the setups involving ground effects. In this case it needs 394 steps to calculate the lower bounds for the coercivity constant. Also the number of basis functions both for the primal and the dual problem is relatively high. The implications are on the one hand an increased offline time due to the costly SCM process. On the other hand the RB online time $t_{RB}^{\text{online},100}$ is high compared to the previous example. Nevertheless, we achieve a speedup of 65 compared to the pure FE solution. It is also worth mentioning that the additional effort for the computation of the dual RB space is only one hour and thus negligible with respect to the primal offline time for the RB method for this setup.

The influence of the ground effect in combination with the Peclet number on the temperature field is illustrated in figure 5.12. The thermal boundary layer on the ground and on the profile becomes thinner and and more separated for higher Peclet numbers. Again, the Peclet number has a much stronger influence on the solution than the geometric parameters.

To study the influence of the new parameter, the distance to the ground, the output value and the associated errors are shown in figure 5.13 as a function of μ_2 for fixed values for the thickness and the Peclet number ($\mu_1 = 14$ and $\mu_3 = 50.05$). Here, the full primal and dual RB space have been used. Also the distance parameter has a nearly linear influence in the examined parameter

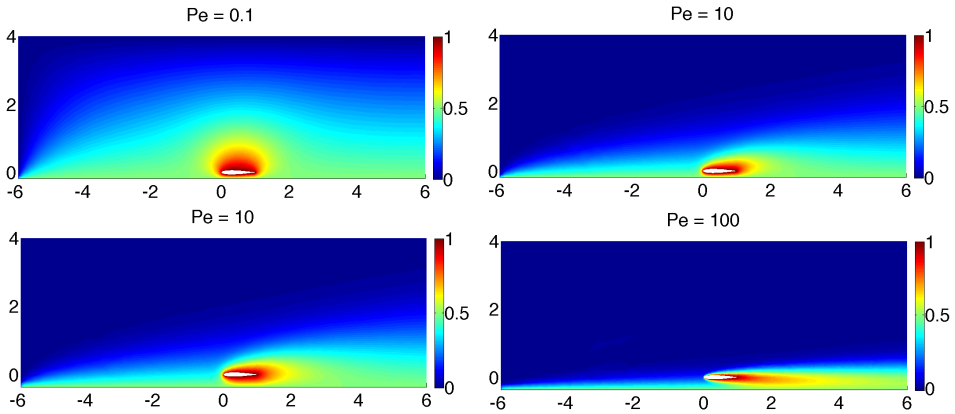


Figure 5.12: Temperature distribution for $\boldsymbol{\mu} = (12, 1.5, 0.1)$, $\boldsymbol{\mu} = (12, 1.5, 10)$, $\boldsymbol{\mu} = (12, 3, 10)$ and $\boldsymbol{\mu} = (12, 3, 100)$ (from upper left to lower right).

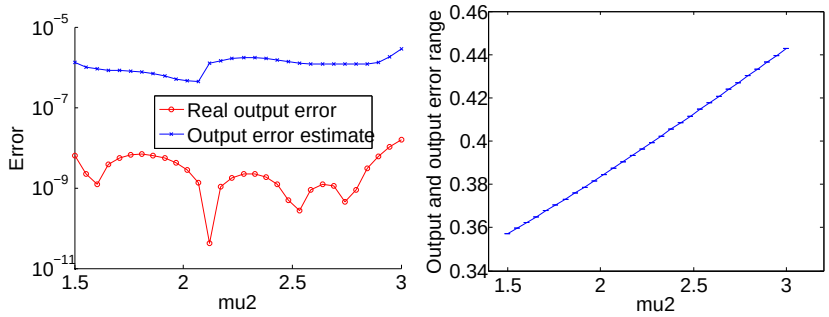


Figure 5.13: Output error and output with error bar interval (based on a posteriori output error estimate) for $\mu_1 = 14$, $\mu_3 = 50.05$, variable μ_2.

range. Again, the error estimates and the real error differ by several orders of magnitude. In contrast to the previous examples, the error estimate follows the qualitative behavior of the real error only roughly but the maximum error is predicted correctly.

Finally, in figures 5.14 and 5.15 the output values and the associated errors for a reduced number of basis functions are shown. Here, the number of primal and dual basis functions has been chosen to be equal. It is interesting to note that the qualitative distribution of the error estimate nearly does not change, but the absolute value of the error estimates decreases by three orders of magnitude when the number of basis functions increases from $N = 15$ to $N = N_{max}$. The real error is decreasing by about the same order of magnitude, but it also changes its qualitative distribution.

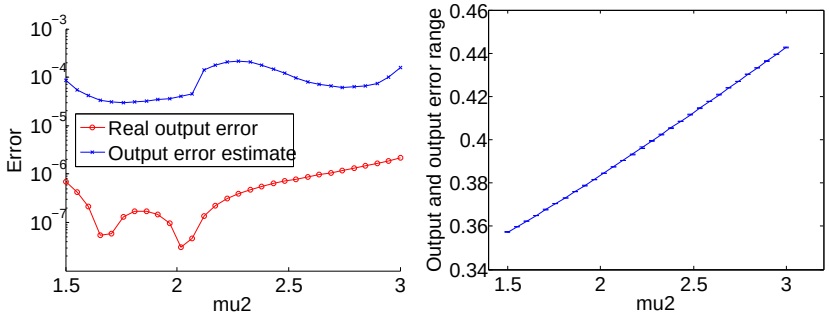


Figure 5.14: Output error and output with error bar interval (based on a posteriori output error estimate) for $N_{pr} = N_{du} = 35$.

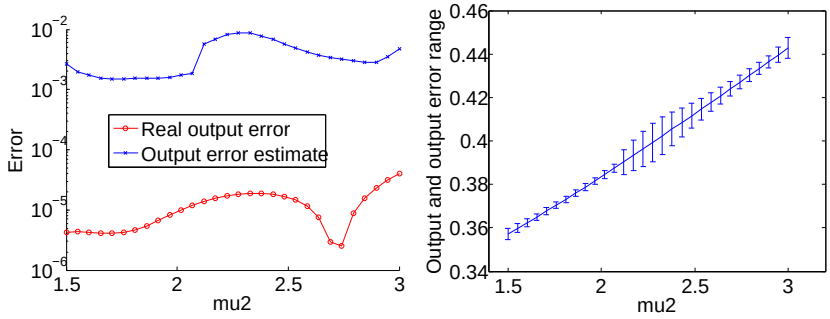


Figure 5.15: Output error and output with error bar interval (based on a posteriori output error estimate) for $N_{pr} = N_{du} = 15$.

5.3.2 Problem 2: Single Profile, with Angle of Attack

Variant A: Angle of Attack Implemented in Boundary Conditions; no Ground Effect

In this variant, the free stream approaches the profile with an angle of attack μ_2 which can be different from zero. The free stream velocity is then set to $\mathbf{w} = (\cos(\mu_2), \sin(\mu_2))^T$. The parameters for this problem are the thickness of the profile $\mu_1 \in [4, 20]$, the angle of attack $\mu_2 \in [0, \pi/4]$ and the Peclet number $\mu_3 \in [0.1, 10]$. Note that the range for the Peclet number has been reduced in this setup to reduce the necessary number of basis functions and thus the computational times both offline and online.

The greedy sample point distributions for the primal and the dual problem are visualized in figure 5.16. The observations of the previous examples concerning the clustering of the sample points on the boundary of the parameter domain are again confirmed. Especially the greedy algorithm for the dual problem chose almost no sample points in the middle of the geometrical parameter domain. In addition, the greedy algorithm for the primal problem shows an interesting accumulation of sample points for small values of μ_1. We note that the clustering in the small and high Peclet number limits seems not as pronounced as the clustering for low and high values in the geometrical parameters.

The influence of the angle of attack in combination with the Peclet number on the temperature field is demonstrated in figure 5.17. While for low Peclet numbers, the angle of attack has almost no influence on the temperature distribution, the influence for higher Peclet numbers where the convection effects are important is very articulated. In addition, it can be observed that for higher Peclet numbers in combination with high angles of attack the chosen zero Dirichlet boundary conditions on the upper domain boundary are not suited anymore. Natural (homogeneous) Neumann conditions would fit better in this case as they would allow the temperature to be convected out of the domain boundaries, as it is the case on the right boundary.

Finally, we study the influence of the angle of attack on the output value and the associated errors in figure 5.18. Here, the output value and errors are depicted as a function of μ_2 for fixed values $\mu_1 = 12$ and $\mu_3 = 5.05$. For this presentation the full primal and dual RB space have been used. Unlike the thickness and the distance from ground, the angle of attack has a nonlinear influence on the output. Of course, for higher angles of attack (when the changes in temperature are interacting not only with the right domain boundary but also with the upper domain boundary), the outputs are not comparable anymore. For lower angles of attack, the influence of the parameter is very weak, while it increases for higher angles of attack. The difference between the real error and the estimated error is two to three orders of magnitude. Compared to problem 1, variant B, where the estimated error did not follow the qualitative behavior of the real error, the qualitative behavior is imitated better for this setup. For higher angles of attack however, the approximation is much better than predicted by the output error estimator.

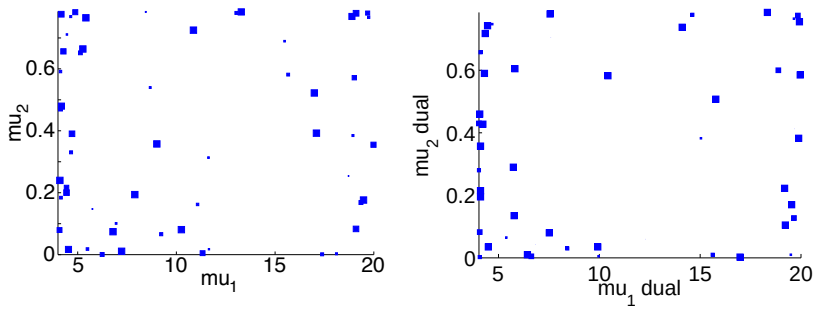


Figure 5.16: Sample points of the primal (left) and dual (right) greedy RB space. Point size is relative to μ_3.

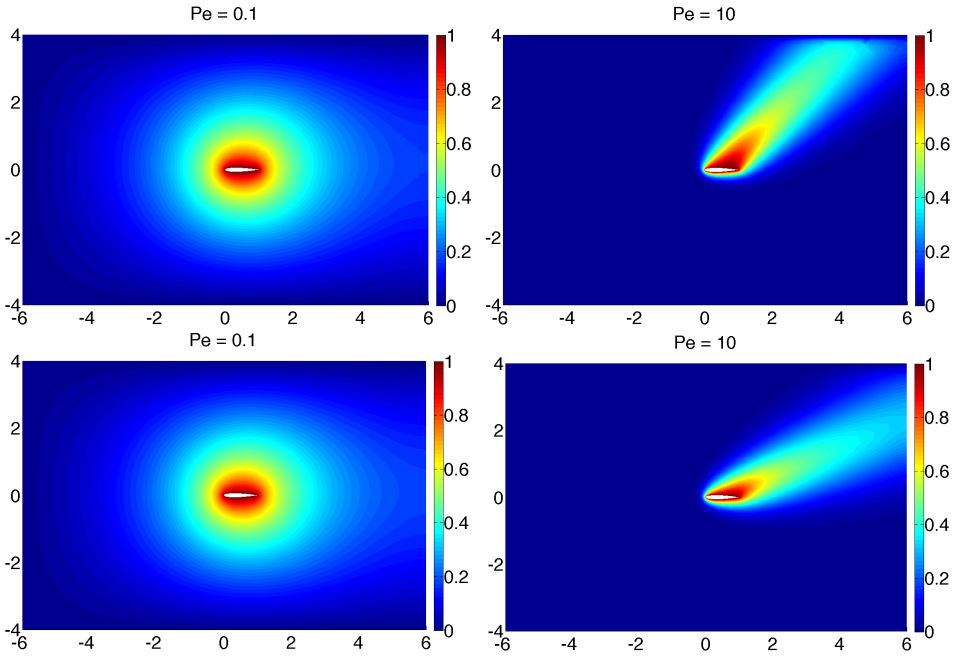


Figure 5.17: Temperature distribution for a thickness of $\mu_1 = 12$ and $\mu_2 = \pi/4\,\mu_3 = 0.1$; $\mu_2 = \pi/4\,\mu_3 = 10$; $\mu_2 = \pi/8$, $\mu_3 = 0.1$; $\mu_2 = \pi/8$, $\mu_3 = 10$; (from upper left to lower right).

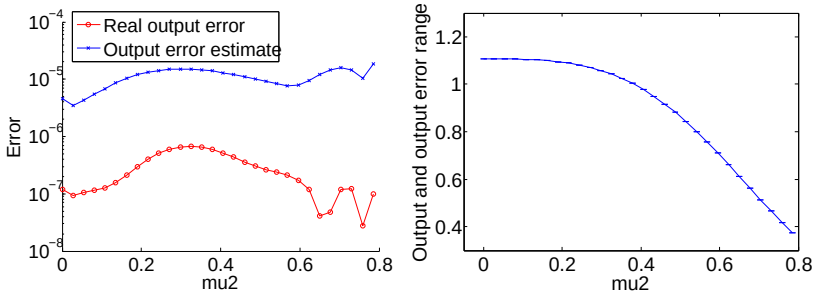


Figure 5.18: Output error and output with error bar interval (based on a posteriori output error estimate) for $\mu_1 = 12$, $\mu_3 = 50.05$, variable μ_2.

Variant B: Angle of Attack Implemented in Boundary Conditions; Ground Effect Included

In the last variant, the free stream approaches the profile with the angle of attack μ_2 and the profile has the distance μ_3 from the ground. In addition, the convection velocity has a linear profile such that it is zero on the ground and directed parallel to the ground. The parameters for this problem are then the thickness of the profile $\mu_1 \in [4, 20]$, the angle of attack $\mu_2 \in [0, 0.15]$, the distance from ground $\mu_3 \in [0.2, 2]$ and the Peclet number $\mu_4 \in [0.1, 10]$.

It is interesting to note from table 5.1 that there are more terms in the bilinear form than subdomains. This can happen in this extent because the convection-diffusion operator consists of two parts, the symmetric and the asymmetric part, and in this case not many of the terms can be combined. In addition, the SCM algorithm needs many steps to fulfill the required quality on the lower bounds for the coercivity constant. We allowed only 200 steps, which results in a value for $\alpha_{tol} \approx 30$. More steps would be needed to get lower bounds of a better quality. However, the overall quality of the estimated error is not extremely worse than in the preceding examples, as we shall see later. Nevertheless, we remark that the RB offline times both for the primal and the dual problem are very high since we used a very fine mesh and the problem consists of many affine terms

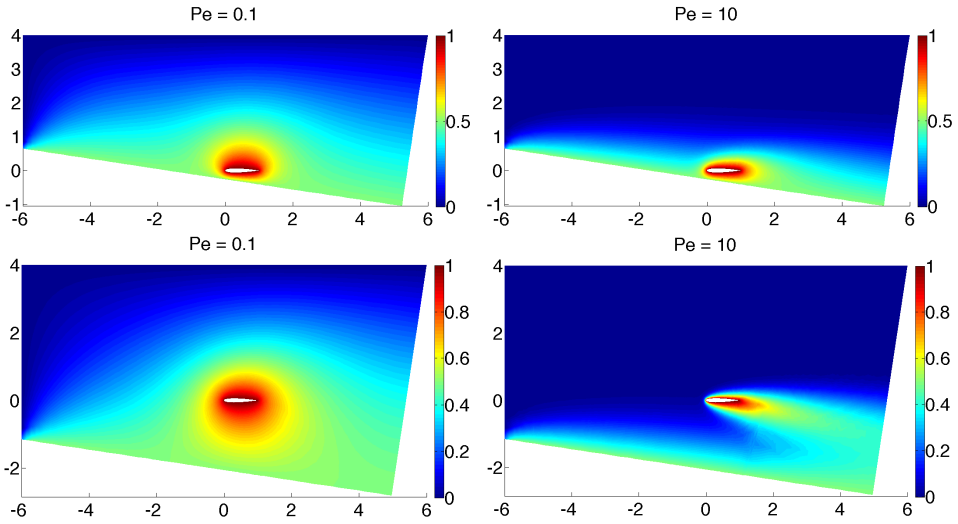


Figure 5.19: Temperature distribution for $\boldsymbol{\mu} = [12, 0.15, 0.2, 0.1]$, $\boldsymbol{\mu} = (12, 0.15, 0.2, 10)$, $\boldsymbol{\mu} = (12, 0.15, 2, 0.1)$ and $\boldsymbol{\mu} = (12, 0.15, 2, 10)$ (from upper left to lower right).

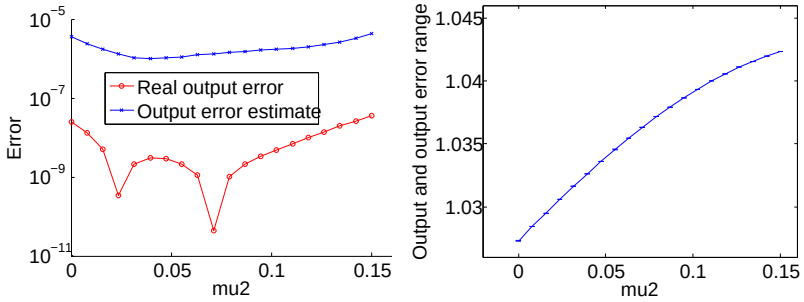


Figure 5.20: Output error and output with error bar interval (based on a posteriori output error estimate) for $\mu_1 = 12$, $\mu_3 = 1.1$, $\mu_4 = 5.05$, variable μ_2.

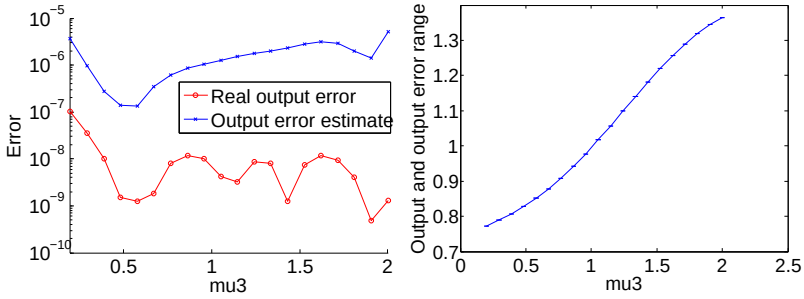


Figure 5.21: Output error and output with error bar interval (based on a posteriori output error estimate) for $\mu_1 = 12$, $\mu_2 = 0.075$, $\mu_4 = 5.05$, variable μ_3.

and basis functions. Especially the SCM algorithm is then very demanding concerning memory and the computational time of the greedy algorithm increases from step to step due to the larger number of basis functions already held in the set.

Figure 5.19 illustrates the behavior of the temperature field for different distances to the ground and Peclet numbers for the maximum allowed angle of attack. Again, the interaction between the two temperature boundary layers can be observed. For low Peclet numbers, the boundary layers are very thick and merge into one single region. For the higher Peclet numbers, especially for higher

	P1VA	P1VB	P2VA	P2VB
# elements mesh N_t	12271	6727	12271	17173
# subdomains	42	88	42	57
affine terms Q_a	24	42	34	90
N_{pr}	31	70	55	112
N_{du}	32	61	50	112
steps SCM $J_{\max}$	8	394	19	200
t_{FE}^{offline}	9min	40min	11min	2h 37min
$t_{RB,pr}^{\text{offline}}$	44min	12h 40min	1h 44min	46h 37min
$t_{RB,du}^{\text{offline}}$	57min	13h 39min	2h 27min	80h 32min
$t_{FE}^{\text{online, 100}}$	38.528s	16.971s	36.921s	113.59s
$t_{RB}^{\text{online,100}}$	0.0436s	0.2612s	0.141s	1.2899s
$t_{RB,out}^{\text{online, 100}}$	0.1544s	0.680s	0.420s	3.46s
speedup	884	65	262	88

Table 5.1: Summary of all setups.

Symbol	Name	Explanation
$t_{RB,pr}^{\text{offline}}$	RB primal offline time	FE offline time + SCM algorithm for $\alpha_{LB}(\boldsymbol{\mu})$ + greedy algorithm for primal RB space
$t_{RB,du}^{\text{offline}}$	RB dual offline time	RB primal offline time + greedy algorithm for dual RB space
$t_{RB,out}^{\text{online,100}}$	RB online time	computation of the scalar output for 100 different parameter values (RB method, full primal & dual space)

Table 5.2: The additional computational times introduced in this chapter.

distances from the ground, the boundary layers are separated on a long part of the domain until they merge further downstream of the profile.

To conclude the presentation of this example, figures 5.20 and 5.21 show the output and output errors for the full primal and dual RB spaces as a function of μ_2 and μ_3, respectively, for fixed values of the remaining parameters. We observe that the angle of attack has now a completely different influence on the output. For higher angles of attack, the output is increasing. Unlike in the separated treatment, the progress is first linear and then becomes flatter. The influence of the distance of the ground is "s-shaped": linear in the middle of the parameter domain, and less steep for low and high values of μ_3. The difference between the estimated and the real error is not increased much compared to the other variants, despite the non-converging SCM algorithm.

Comparison

We will complete this presentation with a comparison of the performance of the RB method for all four setups based on table 5.1. We note that the RB method needs considerably more offline time for the two setups involving ground effects. This is due on the one hand to the more complicated geometry decomposition for the narrow region between the profile and the ground which results in a higher number of affine subdomains and terms for these setups, and on the other hand the SCM method and the greedy algorithm need more steps and more basis functions to fulfill the accuracy requirements. Comparing the RB online times and speedups, again the setups without ground effects perform better since they require less basis functions. In addition, we see that despite the high number of affine terms and basis functions, the setup of problem 2, variant B performs better than the one of problem 1, variant B, since the FE mesh of the underlying problem is finer for the former one.

6 The Reduced Basis Method for Stokes Flow by a Penalty Method

After we illustrated the different steps and aspects of the reduced basis method in the two previous chapters with examples for potential and thermal flows, we will now apply the reduced basis method to creeping flows described by a reformulated Stokes system. The focus of the evaluation for the different geometries described in section 3.2 is laid again on the computation of the scalar outputs and the a posteriori error estimation for the outputs. In addition, the scalar outputs themselves shall be investigated and discussed and an optimization of some of the geometries with respect to lift and drag shall be carried out.

In the next sections, we will first introduce the Stokes problem and the necessary governing equations for the standard Stokes system. We will then reformulate this system by a penalty method to gain a system which is equivalent to the linear elasticity equations and state the weak formulation. After the presentation of the numerical results for the different geometrical setups, we will conclude with a discussion of the reduced basis method applied in the domain of shape optimization. In this context, we will first give a short introduction to shape optimization and constitute the capabilities of the reduced basis method in this domain. Finally, we will present the results of the shape optimization applied to the geometrical setups.

6.1 The Stokes Problem

Stokes flows or "creeping flows" are incompressible flows for which the Reynolds number $Re = LU\rho/\mu$ is very small [53]. Here, L is a characteristic length of the domain, U is a characteristic velocity and ρ and μ are the density and the viscosity of the fluid. The Reynolds number can be small either because the domain or the object is very small, the fluid velocity or the density is very small or the fluid is very viscous. This is the case for many applications, including e.g. blood flow in the human body or groundwater flow in porous media. Of course, in the domain of F1 race car engineering, the Reynolds numbers involved in most of the flow phenomena around the car will be quite high and therefore Stokes flow can be considered only as intermediate or reduced model, see section 3.3.

Stokes flow is governed by the following system of equations: (see e.g. [53], [62])

$$\begin{aligned} -\mu\Delta\boldsymbol{u} + \nabla p &= \boldsymbol{f} \qquad \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 \qquad \text{in } \Omega. \end{aligned} \tag{6.1}$$

The first equation is a simplified momentum equation that results of the full momentum equation in the low Reynolds number limit. The second equation is the continuity equation. We see that we are no longer solving a scalar equation - in the two-dimensional case, we have three unknowns. The two velocity components and the pressure have to be computed. In addition, we have a coupled system of partial differential equations that we have to solve. Details about the FE solution of the Stokes system can be found e.g. in [14] and in [15].

In this case, we made use of a penalty method to approximate the Stokes flow by the use of the linear elasticity equations, see section 6.2. This approach has some important advantages. The main benefit is that we can remain in the coercive case while solving the standard Stokes system, we

would loose this property. This would make the computations involved in the RB solution (mainly offline) a lot more expensive. The main drawback however is that we are not solving the "real" Stokes system, but an approximation of this system which is not fulfilling the continuity equation perfectly.

Although in the domain of F1 race car engineering, in most cases the full Navier-Stokes equations have to be considered and the physical problem can not be approximated well by the model of Stokes flow because the Reynolds numbers involved are too high, Stokes flow is an important intermediate step towards the simulation of the full Navier-Stokes equations. In our case, it adds viscous effects to our problem and thus allows us to calculate the aerodynamic forces that act upon the profiles. We are considering the lift and the drag induced on the profile by the flow. These forces have been subject to many theoretical and practical investigations, mainly in the domain of aircraft engineering. They are crucial for the performance and the capabilities of airplanes and therefore a lot of efforts have been made to reduce the drag while maintaining a given value for the lift. An example for optimum aerodynamic design by the help of CFD techniques is given e.g. in [29]. For high performance cars, lift and drag on wing profiles play also a very important role. In this case, negative lift introduced by attachment parts like wings and flaps is desired to increase the road adherence and to enhance the balance between the forces on the front and the rear axis and thus to allow the pilot to drive through curves with higher speed without loosing control over the car. At the same time, the additional drag introduced by these attachment parts has to be reduced as much as possible to keep the maximum achievable velocity with the given engine power high. An introduction to the aerodynamics involved in race car engineering can be found e.g. in [20].

6.2 Approximation of Stokes Flow by a Penalty Method

In the classical penalty method for the Stokes equation (see e.g. [32]), the continuity equation

$$\nabla \cdot \boldsymbol{u} = 0 \tag{6.2}$$

is replaced by the perturbed equation

$$\nabla \cdot \boldsymbol{u} = -\epsilon p \quad \text{in } \Omega. \tag{6.3}$$

Here, ϵ has to be a small parameter. This allows to eliminate the pressure in the equation system (6.1). The resulting equation system is then

$$-\mu\Delta(\boldsymbol{u_\epsilon}) - \nabla\left(\frac{1}{\epsilon}\nabla \cdot \boldsymbol{u_\epsilon}\right) = \boldsymbol{f} \qquad \text{in } \Omega. \tag{6.4}$$

By following this approach, the size of the system of equations can be reduced. For $\epsilon \to 0$ the solution of (6.4) converges to the solution of (6.1) with $\mathcal{O}(\epsilon)$ [73]. The original continuity equation is no longer fulfilled, thus ϵ has to be small enough to guarantee an accurate solution of (6.1).

Once the solution of (6.4) is computed, the pressure can be obtained from (6.3) by

$$p = -\frac{1}{\epsilon}\nabla \cdot \boldsymbol{u_\epsilon}. \tag{6.5}$$

This means that the solution of an additional linear system is required. The problem with the penalty method is however that for $\epsilon \to 0$, the spectral condition number of the matrix corresponding to the Galerkin FE solution of (6.4) behaves like $1/\epsilon$ (see [21], [62]) and therefore the convergence rate will deteriorate rapidly. The discrete FE solution is said to "lock", which means that the solution remains at very erroneous values without further convergence towards the correct solution as the continuity term $\nabla \cdot \boldsymbol{u} = 0$ is dominating the equations. Techniques to avoid

this are for example the use of an approximate integration for the penalty term. This could be a Gauss quadrature rule of lower order than required as mentioned in [12]. Values for ϵ that have been chosen for physically meaningful numerical calculations in the literature vary approximately in $\epsilon \in [10^{-8}, 10^{-5}]$ (see e.g. [12], [51]). In [21], a value of $\epsilon = 10^{-7}$ has been proposed.

As shown in [10], the resulting penalty system (6.4) is, for homogeneous Dirichlet boundary conditions on the whole boundary $\partial\Omega$, equivalent to the system governing linear elasticity problems:

$$-2\mu \text{ div } \mathcal{E}(\boldsymbol{u}) - \lambda\nabla(\nabla\cdot\boldsymbol{u}) = -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla(\nabla\cdot\boldsymbol{u}) = \boldsymbol{f}. \tag{6.6}$$

Here, $\mathcal{E}(\boldsymbol{u})$ is the linearized strain tensor given by

$$\mathcal{E}_{ij}(\boldsymbol{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{6.7}$$

λ and μ are the so called Lamé constants and $\boldsymbol{u}$ is the displacement variable. This means, that for large λ, we are solving the Stokes problem by a penalty approximation. By using the generalized Hooke's Law (see for example [18])

$$\sigma_{ij}(\boldsymbol{u}) = 2\mu\mathcal{E}_{ij}(\boldsymbol{u}) + \lambda\delta_{ij}\mathcal{E}_{kk}(\boldsymbol{u}), \tag{6.8}$$

where σ is the stress tensor, δ_{ij} is the Kronecker delta and $\mathcal{E}_{kk}(\boldsymbol{u})$ means the trace of $\mathcal{E}$, equation (6.6) can be derived of the general equilibrium equations

$$\frac{\partial\sigma_{ij}}{\partial x_j} + f_i = 0 \qquad \text{in } \Omega. \tag{6.9}$$

For isotropic materials, the components of the stress tensor σ_{ij} can be written as

$$\sigma_{ij} = C_{ijkl}\frac{\partial u_k}{\partial x_l} \tag{6.10}$$

where C_{ijkl} is the elasticity tensor. For the case of linear elasticity and isotropic materials, it has the form (see [30] and [75])

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \tag{6.11}$$

Normally, the Lamé constants are not used but they are replaced by a set of engineering constants: Young's modus E and Poisson's ratio ν. The relations between these constants are given as follows [18]:

$$\mu = \frac{E}{2(1+\nu)} \tag{6.12}$$

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{6.13}$$

$$E = \frac{\mu(2\mu+3\lambda)}{\mu+\lambda} \tag{6.14}$$

$$\nu = \frac{\lambda}{2(\mu+\lambda)}. \tag{6.15}$$

As mentioned before, to approximate Stokes flow by the linear elasticity equations, we have to use $\lambda \to \infty$, which corresponds to an almost incompressible material and the displacement variable $\boldsymbol{u}$ can be interpreted as the material velocity. In terms of the engineering constants, this means that Poisson's ratio ν has to be close to 1/2. For our calculations we used the values $\nu = 0.475$ and $E = 1$, which corresponds to $\lambda \approx 6.44$. This means that we are still far from the limiting case

where the elasticity problem is equivalent to Stokes flow and the continuity equation is not fulfilled very accurately. The result is that we include artificial mass sources and/or sinks in our domain. The reason for this choice of the parameters is the bad numerical conditioning that is provoked by very high values for λ, respectively very small values for ϵ as explained before. As we were using the standard capabilities of the rbMIT package for linear elasticity problems, which do not provide any techniques to prevent the solution from "locking", we had to make sure that the solution is not spoiled by numeric effects. For the chosen values, the solution of the problems remained stable but we had to accept the bad representation of physics. For higher values of λ, respectively ν closer to 1/2, the errors and instabilities arising from the numerical solution of the underlying FE problems grew very fast.

In the two-dimensional case, to approximate Stokes flow, we have to impose the plane strain approximation $\mathcal{E}_{zz} = \mathcal{E}_{yz} = \mathcal{E}_{xz} = 0$. In addition, we assume that all tractions and body forces do not vary with z, that means that they are only functions of x and y.

6.3 Problem Statement

The weak formulation of equation (6.9) is again standard. Multiplying equation (6.9) by the test function $v \in X^e$, integrating on $\Omega_o(\boldsymbol{\mu})$, using integration by parts and the divergence theorem and applying boundary conditions we get

$$\int_{\Omega_o(\boldsymbol{\mu})} \frac{\partial v_i}{\partial x_j} \sigma_{ij} = \int_{\Omega_o(\boldsymbol{\mu})} v_i b_i + \int_{\partial\Omega_o^N(\boldsymbol{\mu})} (v_i f_n e_i^n + v_i f_t e_i^t), \qquad \forall v = (v_1, v_2)^T \in X^e. \tag{6.16}$$

In this formulation, we used the boundary conditions

$$\sigma_{ij} e_j^n = f_n e_i^n \quad \text{on } \partial\Omega_o^N(\boldsymbol{\mu}), \tag{6.17}$$

$$\sigma_{ij} e_j^t = f_t e_i^t \quad \text{on } \partial\Omega_o^N(\boldsymbol{\mu}), \tag{6.18}$$

where e_i^n and e_i^t represent the components of the normal (n) and tangential (t) unit vector.

The compact formulation of our weak statement then follows as

$$a(w, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in X^e, \tag{6.19}$$

where

$$a(w, v; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} \frac{\partial v_i}{\partial x_j} C_{ijkl} \frac{\partial w_k}{\partial x_l} \tag{6.20}$$

$$f(v; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} v_i b_i + \int_{\partial\Omega_o^N(\boldsymbol{\mu})} (v_i f_n e_i^n + v_i f_t e_i^t). \tag{6.21}$$

The formulation on the reference domain Ω (2.1) follows then with the methods of section 2.7.

6.4 Numerical Results

In the following, we will present the numerical results for approximated Stokes flow that we obtained by applying the RB method to the linear elasticity equations for plane strain with a Poisson's ratio of $\nu = 0.475$ and a Young's modulus of $E = 1$. We considered the four different geometry setups that have already been investigated for the thermal flows: the single profile with zero angle of attack without and with ground effects included, and the same setups with the angle of attack included as additional parameter. For these calculations, we did not use the Matlab meshing facilities anymore, but included the meshing routines of COMSOL via the COMSOL interface that

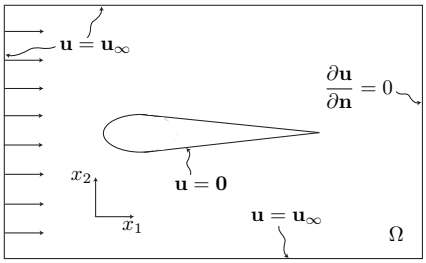


Figure 6.1: Boundary conditions.

the rbMIT code provides. This allowed us to work with meshes with a better quality where the differences in the mesh density are reduced considerably compared to the Matlab-meshes used in the previous chapters.

All problems are again based on the rectangular domain of the size $[-6, 6] \times [-4, 4]$, the big domain. In the setups with ground effect, the lower domain boundary is moved closer to the profile according to the chosen distance from the ground. The boundary conditions are chosen similarly for all four setups, see figure 6.1. On the profile, the velocities have to be zero, so homogeneous Dirichlet conditions in both directions x_1 and x_2 have been applied. On the left, upper and lower domain boundary, the boundary conditions have to impose the free flow. Therefore, we chose inhomogeneous Dirichlet boundary conditions to impose either a flow parallel to the x_1-axis ($u = 1$, $v = 0$) or a flow with a parameter dependent angle of attack ($u = \cos\alpha$, $v = \sin\alpha$). In the variants with ground effect, the model of a profile (eventually turned by the angle of attack) moving parallel to the ground has been adopted ("real" driving situation). That means, that for a correct representation of this situation, the velocity on the "ground" has to be the free stream velocity directed parallel to the ground. The situation that can be found e.g. in a wind-tunnel, where the profile is fixed over the ground and the air is blown past the profile, could be modeled by using homogeneous Dirichlet conditions on the ground and thus imposing zero air velocity on the ground. On the right domain boundary, homogeneous (natural) Neumann outflow conditions have been imposed.

The scalar outputs that we are calculating are the lift and the drag force that act on the profile. They can be calculated as the integral of the stresses $C_{ijkl}(\partial u_k/\partial x_l)n_j$ over the surface of the profile. The component in x_1-direction is interpreted as negative drag, while the component in x_2-direction is interpreted as negative lift, as the actual drag and lift forces are directed in the opposite direction. For the two setups including an angle of attack, the calculated "lift" and "drag" output values are directed in the x_1 and x_2-direction of the fixed coordinate system. In fact, as we are approximating Stokes flow, the domain boundaries are chosen too close to the profile and the results would change considerably if the domain would be chosen bigger. One of the main characteristics of Stokes flow is that the influence of a body on the flow extends very far in all directions and therefore the domain would have to stretch theoretically nearly to infinity (see [53]). If the domain is chosen small (as we did), one has to be aware that the the boundary conditions have an effect on the solution close to the profile and therefore also on the outputs, the lift and the drag force that act on the profile.

As we are working with the penalty method to approximate the Stokes flow, a mass sink is included in the calculations, and therefore the application of compatible Dirichlet boundary conditions on the right domain boundary would not lead to the same result as the application of homogeneous Neumann conditions. The Dirichlet conditions would enforce the same mass flow on the right domain boundary that entered on the corresponding inflow boundaries, while the Neumann conditions allow for a lower mass flow that is determined by the mass sink (which is directly related to our choice of the penalty variable ϵ).

6.4.1 Problem 1: Single Profile, no Angle of Attack

Variant A: No Ground Effect

In this problem, the free stream approaches the profile with zero angle of attack. The single parameter for this problem is the thickness of the profile μ_1. It may vary in $\mu_1 \in [4, 24]$. An overview of the parameters and parameter ranges involved in this and the following problems is given in table 6.2.

The FE meshes for the following problems can be found in appendix A. These meshes have clearly a higher quality than the meshes created by the Matlab meshing facilities and are better suited for the calculation of Stokes flow. For the relevant parameters, such as the the number of affine terms Q_a and the number of mesh elements N_t, refer to table 6.1. Note that the linear form has again zero terms for all four problems that will be presented in this chapter.

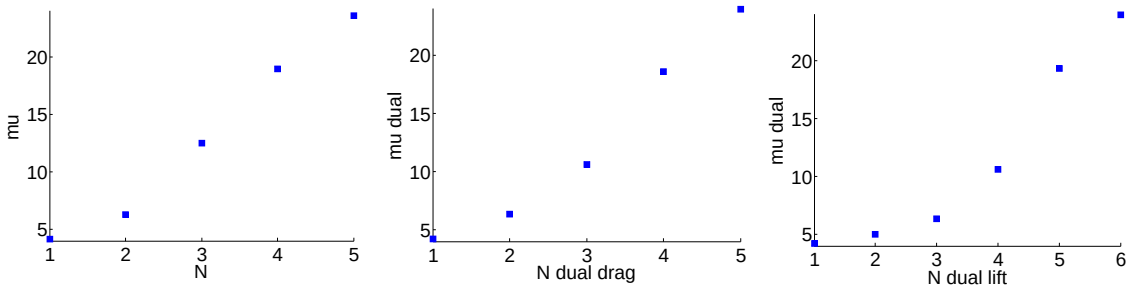


Figure 6.2: Sample points of the primal (first), the dual drag (second) and the dual lift (third) greedy RB spaces.

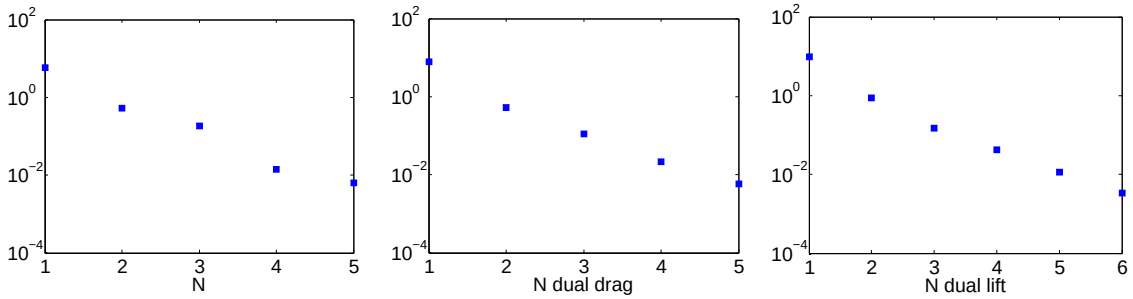


Figure 6.3: Maximum relative energy error as a function of the number of basis functions N for the primal (first), the dual drag (second) and the dual lift (third) problem.

As we need to calculate a different dual RB space for each output, we now work with three different RB spaces. The greedy algorithm needs to choose $N_{pr} = 5$ RB basis functions for the primal RB spaces, $N_{du,\text{drag}} = 5$ RB basis functions for the dual problem related to the drag calculation and $N_{du,\text{lift}} = 6$ RB basis functions for the dual problem related to the lift calculation. The greedy sample point distribution for the primal and the dual problem are visualized in figure 6.2. The clustering of the sample points that could be observed in the previous reports is not very pronounced here. Nevertheless, a slightly higher density for the smaller parameter values can be observed for all three RB spaces. The corresponding maximum relative energy errors (for the field solution) for the primal and the dual problems are shown in figure 6.3. The error reduction rate is comparable for all three problems.

To illustrate the behavior of the solution for this fluid model, the field solutions for the two velocity components, the pressure and a detail of the streamlines around the profile are shown in figure 6.4 for a thickness of $\mu = 24$. First, we note the mass losses due to the use of the penalty

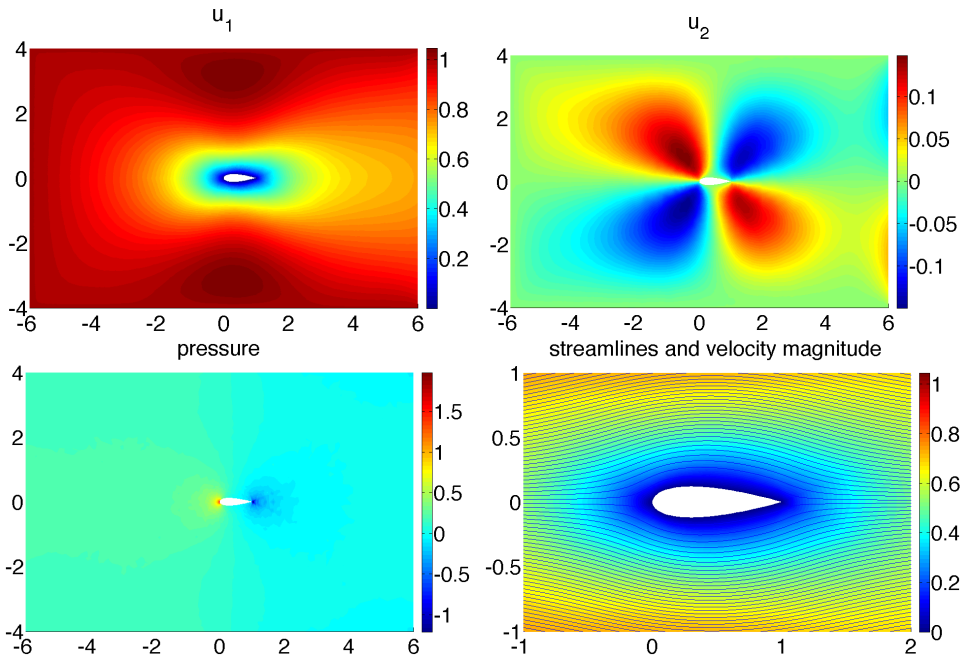


Figure 6.4: Velocity components, pressure field and streamlines together with velocity magnitude for $\mu = 24$.

method. The mass flow over the left domain boundary is higher than the mass flow over the right domain boundary, with no streamlines crossing the upper and lower domain boundary. This causes a stronger asymmetry of the flow in x_1-direction than it would be caused by the slightly asymmetric profile. The pressure has been calculated for visualization purposes by a low order finite difference approximation of the velocity gradients and is therefore subject to numerical errors and has an increased dependence of the mesh quality. Therefore, it is not as smooth as the solutions for the two velocity components. The front and the rear stagnation points are situated precisely at the nose respectively the tail of the profile and the velocity is decreasing correctly in the vicinity of the profile to fulfill the requirement of zero velocity on the surface of the profiles.

We will now concentrate on the output computation. In figure 6.5, the computed values for the negative drag and lift are shown. The negative drag varies linearly with the thickness of the profile and decreases for increasing thickness. That means that the real drag force acting on the profile increases as the thickness is increasing. This is correct according to the experimental observations for the NACA-profiles [1]. The lift should be - in theory - zero. Due to slight asymmetries in the mesh and therefore also in the solution, it is not exactly zero but very small. For thicker profiles, the influence of the asymmetry seems to decrease and so the negative lift is closer to zero for thicker profiles.

Figure 6.6 shows the variation of the error in the outputs with the thickness of the profiles. It can be observed that the output error estimates are slightly sharper than for the thermal flows studied in chapter 5 in this case, especially for the drag. Nevertheless, the real error in the output values is still two to three orders of magnitude smaller than the estimated error, even if this is due to the security margins provided by error bounds. The qualitative behavior of the error is represented extremely well by the error estimator in this case.

Figure 6.7 shows the development of the dual error estimate for an increasing number of RB basis functions (from zero to $N_{\max,du}$) in the dual RB space for the drag and lift outputs together with the error estimate for the output error if only the primal approximation is used. Each blue line represents a different number of dual basis functions, the top line is the error estimate if no dual basis functions are used, the lowest line is the error estimate if all available dual basis functions

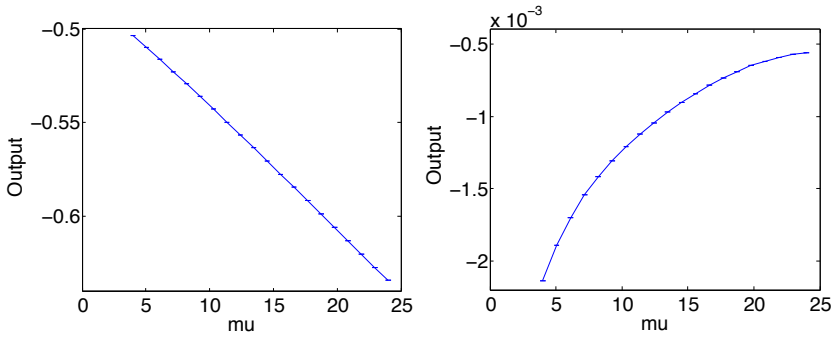


Figure 6.5: Variation of the output functionals over the parameter domain together with errorbar intervals for drag (left) and lift (right).

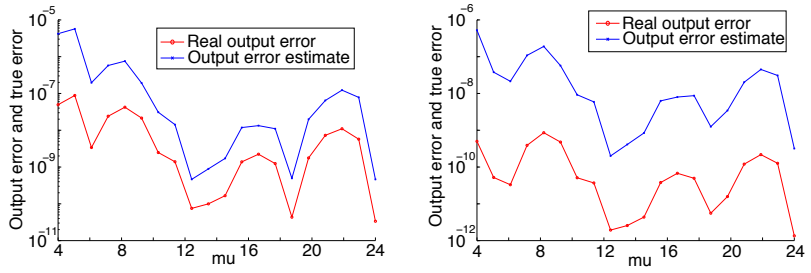


Figure 6.6: Comparison of the output error estimates with the real output error for drag (left) and lift (right).

are used. The reduction of the dual error estimate with each new discrete sample point can be observed quite well. The estimated error is smaller on the whole parameter domain if a new basis function is added, and for the chosen sample point and its vicinity, it decreases considerably. Like this, discrete peaks are created in the vicinity of the parameter sample points, where the error is much lower than for the rest of the domain. It has to be noted that if no dual basis functions are used, the error estimator for the dual error does not produce the same result as the error estimator for the primal error, because slightly different error estimators are used. The dual error estimate actually makes sense only if there are indeed dual basis functions used in the solution process.

To conclude the discussion of this first setup, we will give an impression of the performance of the RB method for this fluid model. For this geometry setup, the FE offline time has been $t_{FE}^{\text{offline}} = 43min$. The increased time compared to the solution of e.g. the convection-diffusion

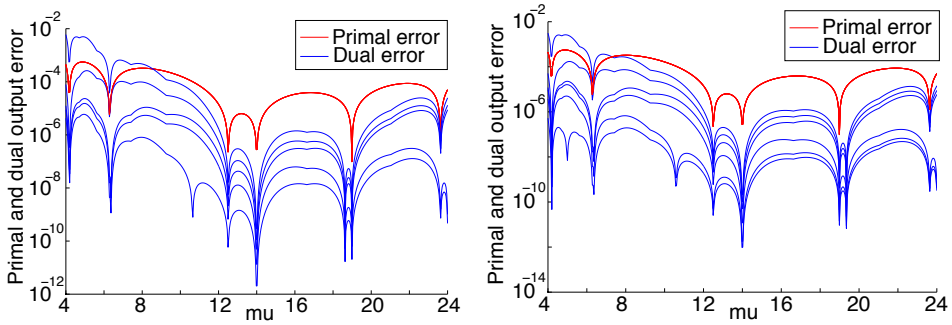


Figure 6.7: A posteriori output error estimates for primal and dual error for the computation of the drag (left) and the lift (right); the number of dual basis functions increases from the top to the lowest blue line from zero to $N_{\max,du}$.

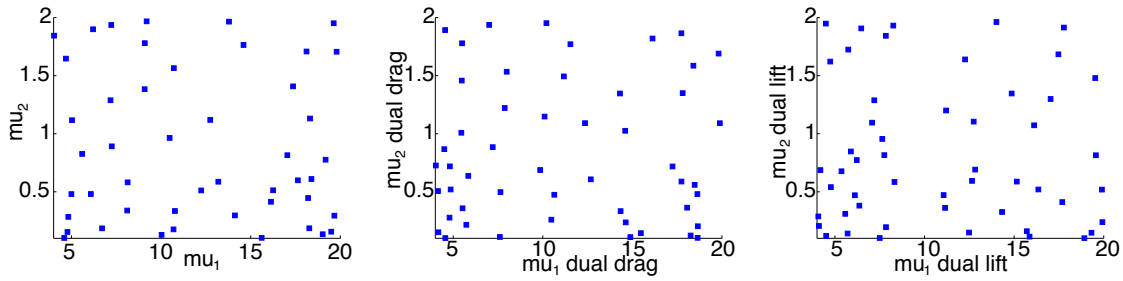


Figure 6.8: Sample points of the primal (first), the dual drag (second) and the dual lift (third) greedy RB spaces.

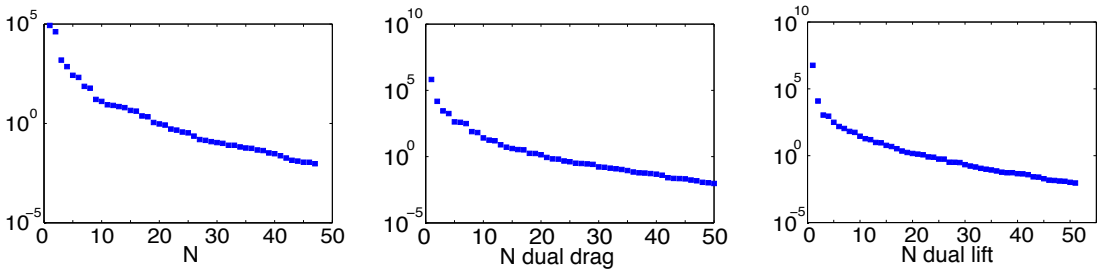


Figure 6.9: Maximum relative energy error as a function of the number of basis functions N for the primal (first), the dual drag (second) and the dual lift (third) problem.

equation with a FE mesh with a comparable number of mesh elements is due to the fact that we are now solving a system of equations and not a scalar problem anymore. The primal offline time for the RB method (defined as in chapter 5) has been $t_{RB,pr}^{\text{offline}} = 1h\,14min$ and thus the additional effort compared to the FE method is very small. The dual offline time for the RB method has been $t_{RB,du}^{\text{offline}} = 1h\,19min$. The additional time required to form the dual approximation spaces can thus be neglected in this case. The FE online time for the computation of 100 field solutions is $t_{FE}^{\text{online, }100} = 29.394s$, while the time for the RB method is $t_{RB}^{\text{online, }100} = 2.34 \cdot 10^{-2}s$ and thus it is about 1250 times faster. This outstanding performance is due to the very fine mesh and the small number of basis functions and affine terms in the bilinear form. In addition, we measured the time to compute the scalar outputs for 100 parameter instances with the full primal and dual RB space. In this case, this is $t_{RB,\text{drag}}^{\text{online, }100} = 0.15s$ for the drag computation and $t_{RB,\text{lift}}^{\text{online, }100} = 0.16s$ for the lift computation.

Variant B: Ground Effect Included

This problem combines the setup of variant A with the ground effect. The lower domain boundary is therefore moved to a line parallel to the x_1-axis through $y_L = -\mu_1/200 - \mu_2$, where the parameters are defined according to table 6.2.

As for the previous fluid models, the SCM computation of the lower bounds for the coercivity constant is a lot more demanding for the geometries including ground effects. The SCM algorithm needs 400 steps to get upper and lower bounds that result in $\alpha_{\text{tol}} \approx 3$. More steps would be needed to fulfill the usual requirement $\alpha_{\text{tol}} \leq 0.85$, but the additional accuracy gained by continuing the procedure would not justify the additional offline costs. The RB offline time is already very high due to the high number of SCM steps, see table 6.1. The greedy algorithm needs then to choose $N_{pr} = 47$ RB basis functions for the primal, $N_{du,\text{drag}} = 50$ RB basis functions for the dual problem related to the drag calculation and $N_{du,\text{lift}} = 51$ RB basis functions for the dual problem related to the lift calculation. The greedy sample point distributions for the primal and the dual problem

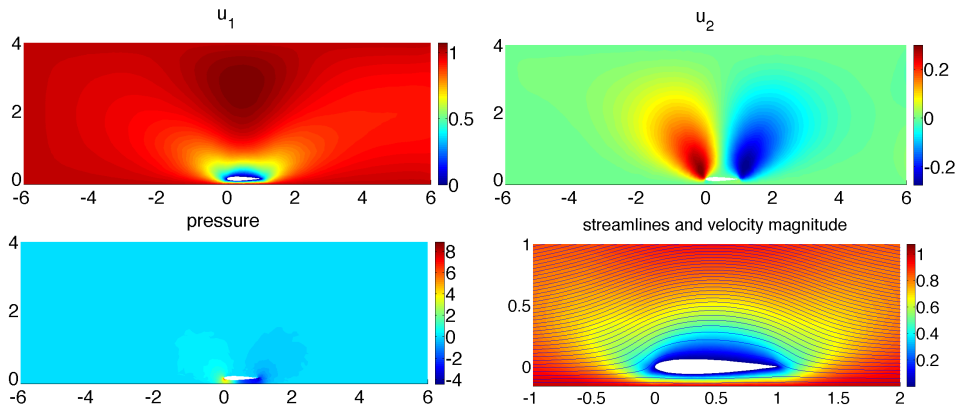


Figure 6.10: Velocity components, pressure field and streamlines together with velocity magnitude for $\boldsymbol{\mu} = (12, 0.1)$.

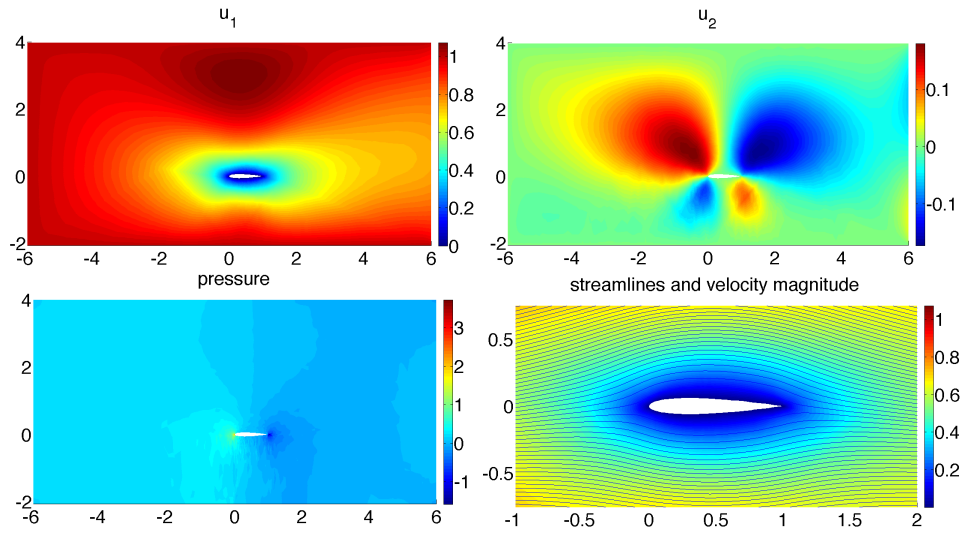


Figure 6.11: Velocity components, pressure field and streamlines together with velocity magnitude for $\boldsymbol{\mu} = (12, 2)$.

are visualized in figure 6.8. Again, the clustering of the sample points on the boundary of the parameter domain that could be observed in the previous reports is not very pronounced here, but an interesting clustering of the sample points for the dual drag basis appears: The density of the sample points for the lower values of μ_2 is increased for nearly discrete values of μ_1. For higher values of μ_2, this effect can not be observed anymore. The corresponding maximum relative energy errors (for the field solution) for the primal and the dual problems are shown in figure 6.9. Note that the error reduction rates for the two different dual problems are almost equal, although there are many differences in the distributions of the sample points.

To illustrate the influence of the ground effect on the solution, the field solutions for the two velocity components, the pressure and a a detail of the streamlines around the profile are shown in figure 6.10 for $\boldsymbol{\mu} = (12, 0.1)$ and in figure 6.11 for $\boldsymbol{\mu} = (12, 2)$. For decreasing distance of the ground, the pressure maximum respectively minimum gets stronger on the lower side of the profile. The flow around the profile and the pressure distribution gets more and more asymmetric, as the flow below the profile is forced to be parallel to the ground. The velocity boundary layer between the profile and the ground is therefore much thinner than the boundary layer on the upper side of the profile for small distances to the ground. If the profile is situated farther away of the ground,

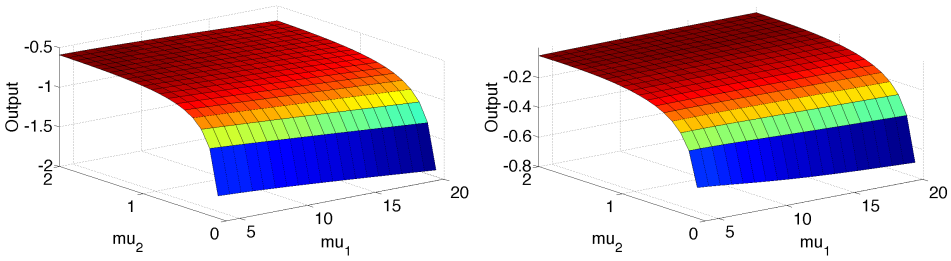


Figure 6.12: Variation of the output functionals over the parameter domain for drag (left) and lift (right).

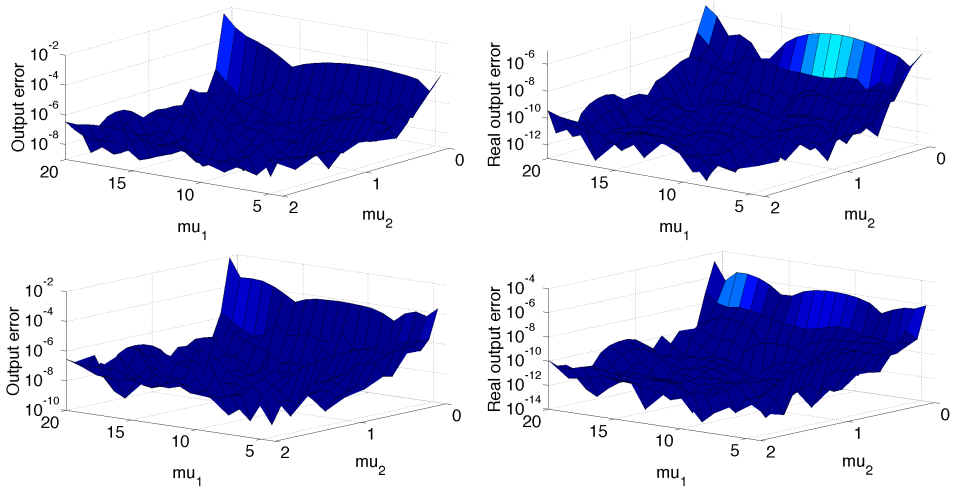


Figure 6.13: Comparison of the output error estimates (left) with the real output error (right) for drag (upper) and lift (lower).

the flow profile gets again more symmetric, but the influence of the ground can still be observed clearly for the maximum distance allowed for this setup, as it is typical for Stokes-flow.

In the following, we will again concentrate on the output computation. In figure 6.12, the computed values for the negative drag and lift are shown. It is obvious that the influence of the distance to the ground is stronger than the influence of the thickness of the profiles. The variation of the output with the distance of the ground is both for negative drag and lift still relatively small for higher values of the distance. If the distance gets smaller, at about $\mu_2 \approx 0.7$, the variation of both outputs starts to get steeper, until they decrease very rapidly for very small distances. While the negative drag decreases with increasing thickness (as already seen for the previous variant) over the whole range for μ_2, the negative lift is nearly independent of the thickness for high values for μ_2 and decreases for increasing thickness for small values of μ_2.

The comparison of the output error estimates with the real output error in figure 6.13 shows that the highest errors occur for small values of μ_2. For both the lift and the drag, the variation of the error over the parameter domain is about five orders of magnitude both for the real error and for the estimated error and thus relatively high. The difference between the estimated error and the real error is about three orders of magnitude.

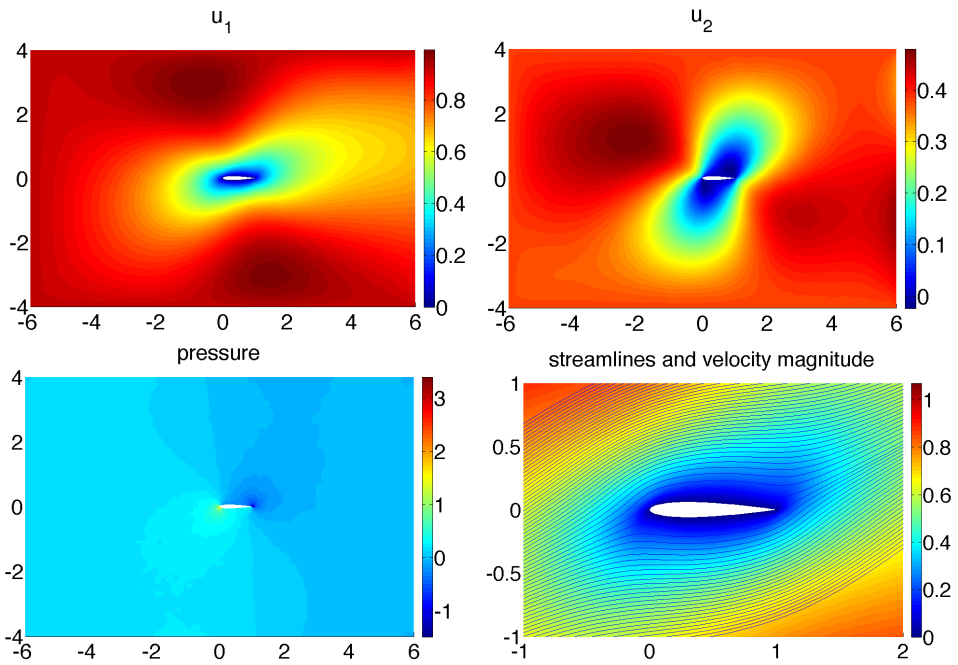


Figure 6.14: Velocity components, pressure field and streamlines together with velocity magnitude for $\boldsymbol{\mu} = (12, \pi/8)$.

6.4.2 Problem 2: Single Profile, with Angle of Attack

Variant A: Angle of Attack Implemented in Boundary Conditions; no Ground Effect

This problem combines the setup of problem 1, variant A with a variable angle of attack implemented in the boundary conditions, see table 6.2. Figure 6.14 shows the field solutions for the velocity components, the pressure and a detail of the streamlines around the profile for $\boldsymbol{\mu} = (12, \pi/8)$. With increasing angle of attack, the region of low velocity around the profile rotates and extends over a bigger domain. The pressure distribution gets more and more asymmetric. The regions of low and high pressure are turned according to the angle of attack, thus the pressure distribution is - as we have seen for potential flows in chapter 4 - somewhat antisymmetric. Contrary to potential flows, Stokes flow can not flow around the sharp trailing edge and thus the rear stagnation point is situated on the trailing edge and not on the upper side of the airfoil. Nevertheless, due to the character of Stokes flow and the chosen boundary conditions, the profile does not change the direction of the flow further downstream, except of a small region in the direct vicinity of the profile.

We will now again discuss the output computation. In figure 6.15, the computed values for the negative drag and lift are shown. Here it has to be kept in mind that the force components are given in the x_1- and x_2- direction of the fixed coordinate system and not in the components parallel and perpendicular to the angle of attack. We see that the absolute value of the negative drag decreases for increasing angle of attack as well as for increasing thickness. The dependence on the thickness seems to be linear, while the steepness of the output functional increases with increasing angle of attack. The negative lift in this coordinate system does not depend on the thickness at all, and its absolute value increases for increasing angles of attack.

The output error estimates for lift and drag are shown in figure 6.16. It is interesting to note that the error does almost not depend on the angle of attack, but it shows negative peaks for some discrete values for the thickness where it is extraordinary small. This behavior indicates that the second parameter has not a big influence on the complexity of the output computation and the error behaves essentially as for problem 1, variant A.

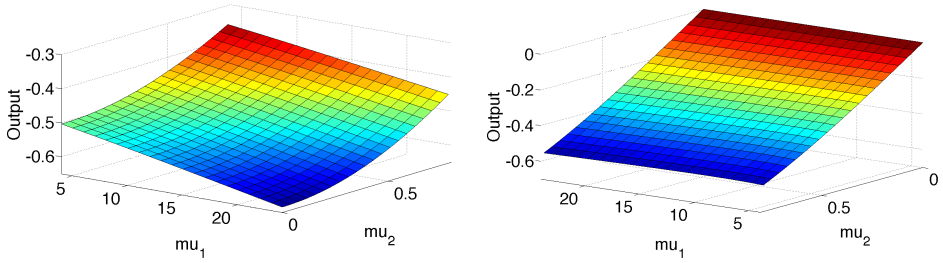


Figure 6.15: Variation of the output functionals over the parameter domain for drag (left) and lift (right).

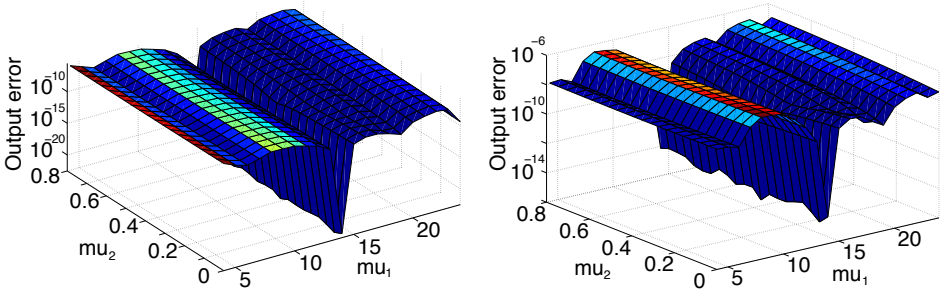


Figure 6.16: Output error estimates for drag (left) and lift (right).

Variant B: Angle of Attack Implemented in Boundary Conditions; Ground Effect Included

Finally, this problem combines the setup of problem 1, variant B with a variable angle of attack implemented in the boundary conditions and in a rotation of the lower domain boundary ("the ground"). To illustrate the combined effects of the angle of attack and the ground effect on the solution, the field solutions for the two velocity components, the pressure and a detail of the streamlines around the profile are shown in figure 6.17 for $\boldsymbol{\mu} = (12, 0.15, 0.2)$. One can see that there is in fact a superposition of the two effects explained before.

To conclude, the behavior of the outputs shall be discussed. In figure 6.18, the computed values for the negative drag are shown as a function of μ_2 and μ_3 for four different values of μ_1. The distance from the ground has the biggest influence on the solution, while again the dependence

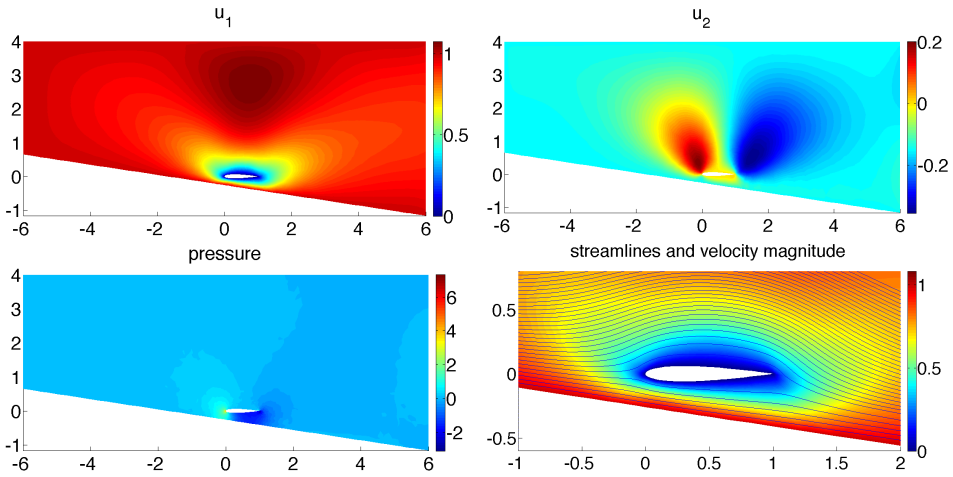


Figure 6.17: Velocity components, pressure field and streamlines together with velocity magnitude for $\boldsymbol{\mu} = (12, 0.15, 0.2)$.

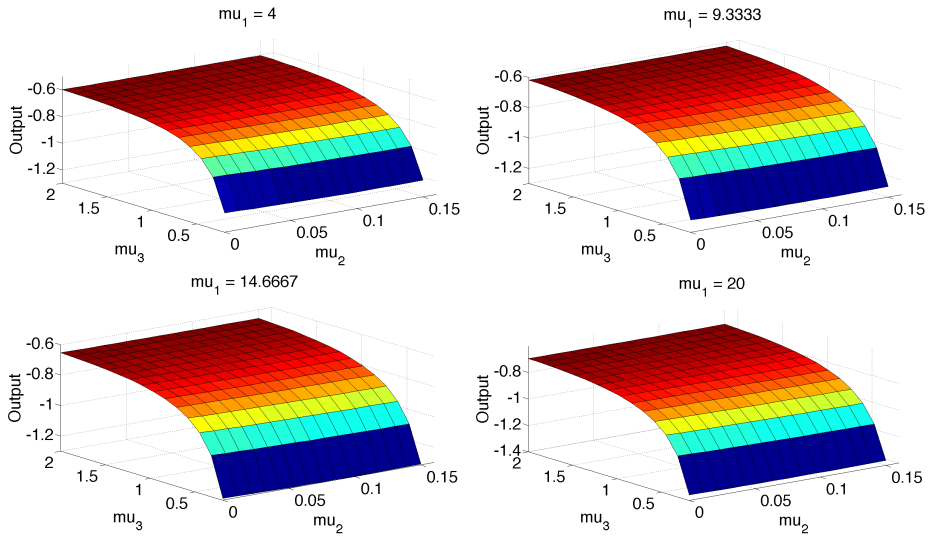


Figure 6.18: Variation of the output functionals over the parameter domain for the drag, μ_1 is increasing from the upper left to the lower right.

on the distance remains relatively weak for higher distances of the ground and starts to get very strong for small distances. This behavior is essentially the same as discussed for problem 1, variant B. The absolute value of the negative drag increases slightly for increasing angle of attack and also for increasing thickness of the profile. The dependence of the negative lift on the three parameters is more complicated, see figure 6.19. While for higher distances of the ground the dependence of the angle of attack is almost linear with a constant gradient, the gradient starts to increase with decreasing distance starting at about $\mu_3 = 0.7$. The dependence of the thickness is not very strong in the limit of higher distances from the ground, but it gets more pronounced for small distances from the ground. The superposition of the effects causes the lift to change the sign for an angle of attack depending on the distance of the ground as the two parameters have an opposed effect on the lift. While the negative lift (the force that is pressing the profile to the ground) increases for increasing negative angle of attack, a decreasing distance from the ground increases the positive lift (the force that is lifting the profile away from the ground).

Comparison

We will complete the discussion of the single setups with a comparison of the performance of the RB method for all four setups based on table 6.1. Comparing the affine decompositions and the number of affine terms, it becomes clear that the two setups without ground effects behave very similar. The setups with ground effects require a much more complicated decomposition and thus result in a higher number of affine terms. The same behavior can be observed in the number of basis functions needed both for the primal and the dual RB spaces. While the setups without ground effect require only a very moderate number of basis functions (which is almost equal for the dual problems), the setups with ground effects need on the one hand a lot more SCM steps, and on the other hand a strongly increased number of basis functions for both the primal and the dual problems. This has of course effects on the offline and online effort, as the offline effort is extremely high for the two setups with ground effects due to the many SCM steps (which account for the biggest part of the offline RB time) and the long greedy sampling procedures. The increased number of basis functions then leads to a lower speedup for these setups compared to the excellent speedup for the two setups of variant A.

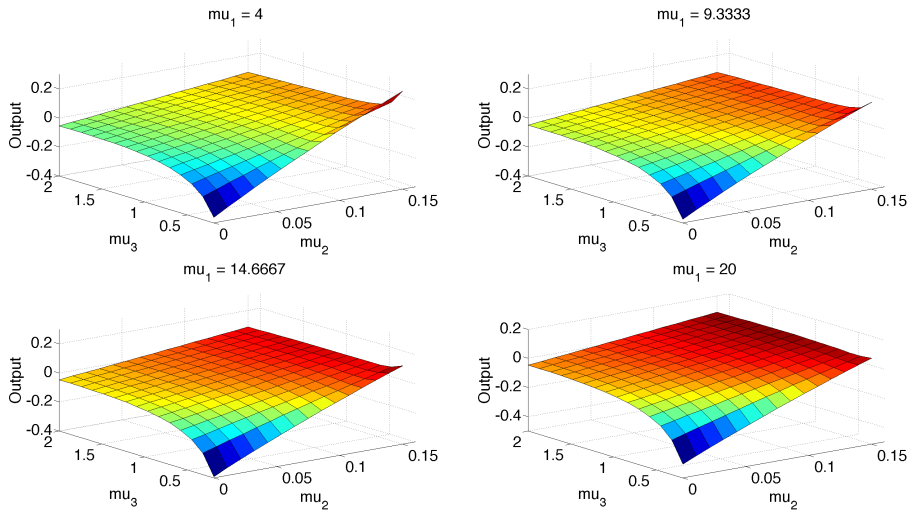


Figure 6.19: Variation of the output functionals over the parameter domain for the lift, μ_1 is increasing from the upper left to the lower right.

	P1VA	P1VB	P2VA	P2VB
# elements mesh N_t	10210	7714	10210	7714
# subdomains	98	99	98	99
affine terms Q_a	30	72	30	82
N_{pr}	5	47	10	63
$N_{du,\text{drag}}$	5	50	5	53
$N_{du,\text{lift}}$	6	52	5	58
steps SCM $J_{\max}$	9	400	9	500
t_{FE}^{offline}	43min	1h 4min	57min	2h 20min
$t_{RB,pr}^{\text{offline}}$	1h 14min	33h 35min	1h 34min	57h 12min
$t_{RB,du}^{\text{offline}}$	1h 19min	39h 43min	1h 40min	66h 56min
$t_{FE}^{\text{online, 100}}$	29.394s	33.785s	32.344s	38.441s
$t_{RB}^{\text{online,100}}$	0.0234s	0.2531s	0.0374s	0.461s
$t_{RB,\text{drag}}^{\text{online, 100}}$	0.15s	0.99s	0.18s	1.38s
$t_{RB,\text{lift}}^{\text{online, 100}}$	0.16s	0.99s	0.18s	1.59s
speedup	1256.16	133.48	864.82	83.39

Table 6.1: Summary of all setups.

Problem	μ_1	Range μ_1	μ_2	Range μ_2	μ_3	Range μ_3
P1 VA	thickness	$[4, 24]$				
P1 VB	thickness	$[4, 20]$	distance to ground	$[0.1, 2]$		
P2 VA	thickness	$[4, 24]$	angle of attack	$[0, \pi/4]$		
P2 VB	thickness	$[4, 20]$	angle of attack	$[0, 0.15]$	distance to ground	$[0.2, 2]$

Table 6.2: Parameters and parameter ranges.

6.5 Shape Optimization

In this section, we will perform the optimization of some of the setups previously discussed with regard to the aerodynamic forces lift and drag. As already mentioned, in the domain of race car engineering, the intention of the use of additional parts such as wings on high performance cars is to add negative lift to the vehicle that increases the maneuverability for high velocities. The additional drag introduced to the system of car and wing should be as small as possible. There are now two different approaches possible, either one tries to maximize the lift-to-drag ratio, or one aims at minimum drag for a prescribed value or allowed range for the lift. The first could be realized by so called "unconstrained optimization", while the second approach is subject to the "constrained optimization" algorithms. In the following, we will very briefly give the basics of shape optimization with respect to the special problems that we are working with and want to highlight the special abilities of the reduced basis method in optimization applications. We will not go into details, as this work focuses on the methodology of the reduced basis method that itself gives us all necessary ingredients for a rapid and reliable optimization with a wide range of possible optimization algorithms.

6.5.1 Introduction

Shape optimization is an example for a special class of optimization problems. It aims at the optimization of one or more cost functionals $J(\boldsymbol{\mu})_i$ which depend on design parameters $\boldsymbol{\mu} \in D$ which define the shape of the system within the admissible set of values D [39]. In our case, the cost functional is depending on the design parameters not only directly, but also through the field variable $\boldsymbol{u}(\boldsymbol{\mu})$ that is the solution of a parametrized partial differential equation. So in fact, our problem may belong to the domain of optimal control, where the solution of a partial differential equation (the state variable) shall be controlled by a parameter or control variable in an optimal way. In addition, our optimization problem can include one or more constraints, that can also be dependent of the parameters $\boldsymbol{\mu}$ and the field variable $\boldsymbol{u}(\boldsymbol{\mu})$. The solution of the optimization problem normally involves the solution of the partial differential equation for many parameter values and often information about the Jacobian or Hessian matrices of the system is needed, too. Therefore, the solution of shape optimization problems usually is very expensive and the use of the reduced basis method in this context to reduce the computational times needed is a very promising approach.

There exist a huge number of different ways and techniques to solve such shape optimization problems. Basically, there can be found two different approaches. The first approach is first to establish all equations necessary to solve the optimization problem and then to discretize the resulting system of partial differential equations and algebraic equations ("optimize then discretize"), while the second approach at first uses suitable discretization strategies for the partial differential equation and/or the cost functionals and constraints and then applies an optimization procedure to the discretized equation systems ("discretize then optimize"). A presentation of general algorithms for optimization, from the well-known general gradient and Newton methods to more specialized and sophisticated procedures, can be found e.g. in [42]. Optimal control for systems governed by partial differential equations is addressed in [33] and a good review of techniques for shape optimization for fluids is presented in [39].

6.5.2 The Reduced Basis Method for Shape Optimization

There are a number of works that used the reduced basis approach for optimization problems following the first approach ("optimize then discretize") (e.g. [67], [61]), but here we want to present the abilities of the reduced basis method in the context of the second approach ("discretize then optimize") for affinely parametrized coercive elliptic partial differential equations. Earlier work

on the application of the reduced basis method in optimization and optimal control for fluid flow problems has been done by Ito and Ravindran ([25], [26] and [27]).

If the cost functional $J(\boldsymbol{\mu})$ is composed only of bounded linear functionals $l(\boldsymbol{u}(\boldsymbol{\mu}))$ that can be computed as outputs of the reduced basis method applied to the governing partial differential equation (and possibly of functions that are directly dependent on the parameters), the RB method is directly applicable for the computation of the cost functional. The computational costs of the output computation by the reduced basis method are much lower than the computational costs that would arise by the computation of the outputs by the use of the FE method, as shown and explained in the previous chapters. But not only the computation of the output values (and thus the cost functional) can benefit from the RB method, but also the computation of the gradients and the Hessian of the outputs can be performed by the use of the RB method with a computational time comparable to that for the computation of the outputs, as demonstrated in [52]. In addition, the RB method provides sharp and rigorous error bounds. We can use these error bounds to guarantee that the constraints on our optimization problem are indeed respected, see section 2.4.5. The sharpness of the error bounds is important to ensure that we do not include security margins that are unnecessarily high and therefore prevent us from finding the true optimal solution. In [52], the authors also show a way how to compute the gradients and Hessians of the Reduced-Basis output error bounds at approximately the same computational cost as for the output computation. The solution of the optimization problem is thus not bound to a specific optimization algorithm, but a procedure that is making use of all this additional information on the output is certainly best suited for this approach.

In our case, we composed the cost functionals of the drag force, the lift force and the thickness of the profile as a measure for its volume, which should be as small as possible. In each case, we considered a constrained problem and an unconstrained problem. The concrete formulation for the cost functionals and the constraints varies for the different setups, so we will mention them for each of the problems below. For the solution of the optimization problem we chose a "brute-force" approach and computed the output values for the complete parameter domain with a very small step size for the different parameters. After that, the "real" aerodynamic forces were calculated of the output functionals by rotating them in the right coordinate system and changing the sign. In this way, we can reproduce the output functionals and aerodynamic forces and search for the minimum of the cost functional (while fulfilling the constraints) afterwards. As the computational times for the output evaluation are very short, this is even feasible up to a moderate number of parameters. If many parameters are included, of course a more sophisticated optimization algorithm should be used.

6.5.3 Numerical Results

Problem 1, Variant B

The first setup we are considering for optimization is problem 1, variant B. As we do not include an angle of attack in our calculations, the drag and lift forces are already given in the right coordinate system and have only to be multiplied by -1 to give the correct lift and drag forces. Figure 6.20 shows the drag and lift defined in the correct directions. As already discussed in the previous sections, both forces are increasing strongly when the profile is approaching the ground. For attachment parts for race cars, this effect is of course undesired both for the drag and the lift and small distances to the ground should therefore be avoided. For real flow situations around race cars where the profiles are also interacting with the rest of the car in a complex way, the effects could be different and there could be reasons to place the attachment parts close to the ground anyways.

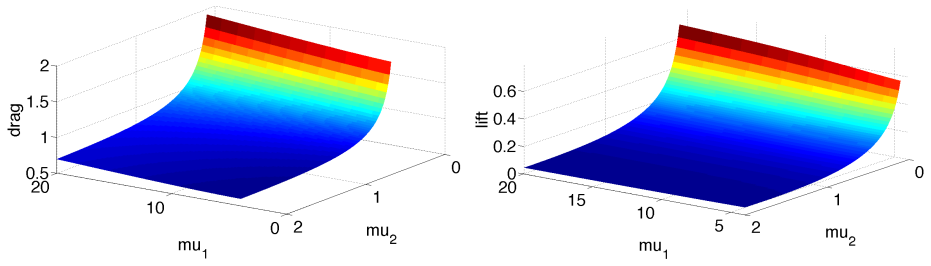


Figure 6.20: Drag (left) and lift (right) forces in correct direction, problem 1, variant B.

	constrained	unconstrained
μ_1^{opt}	4	4
μ_2^{opt}	2	1.8465
D	0.5978	0.5997
L	0.0547	0.0541

Table 6.3: Optimization results, problem 1, variant B.

The constrained optimization has been done by using the following objective function:

$$J(\boldsymbol{\mu}) = 10\frac{D(\boldsymbol{\mu})}{D_{\text{ref}}} + \frac{\boldsymbol{\mu}_1}{V_{\text{ref}}} \tag{6.22}$$

were D is the drag force, D_{ref} is the reference drag force and is given as $D_{\text{ref}} = 0.6$ and $V_{\text{ref}} = 14$ is the reference thickness. These reference values will remain the same for the following setups. The optimization of the drag is weighted stronger than the reduction of the volume. The constraints are given for the lift force. As we have seen that the lift is always positive (and positive lift is normally unwanted for race car attachment parts), a maximum allowed lift force is given as $L_{\max} = 0.2$.

The unconstrained optimization is done with the following objective function:

$$J(\boldsymbol{\mu}) = 10D(\boldsymbol{\mu})L(\boldsymbol{\mu}) + \mu_1. \tag{6.23}$$

Here, the first term tries to minimize both drag and (in this case always positive thus unwanted) lift force. The second term again tries to keep the volume of the profile as small as possible. The results for the two optimization strategies and objective functions are shown in table 6.3. The results are relatively similar as both variants choose the thinnest possible profile in combination with a big distance to the ground. Figure 6.21 shows the resulting pressure distributions for the optimal setups for both optimization variants and cost functionals.

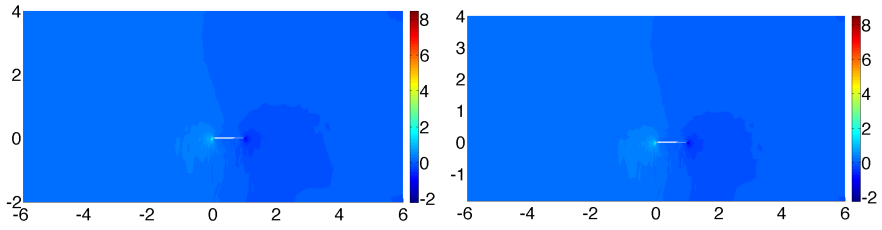


Figure 6.21: Optimal pressure distribution: constrained (left) and unconstrained (right) optimization procedure, problem 1, variant B.

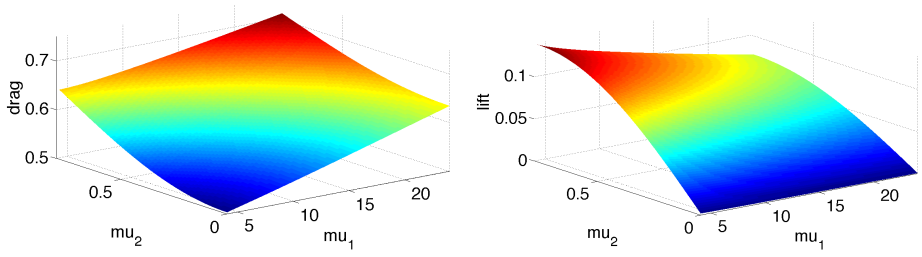


Figure 6.22: Drag (left) and lift (right) forces in correct direction, problem 2, variant A.

	constrained	unconstrained
μ_1^{opt}	4	4
μ_2^{opt}	0.4601	0.6743
D	0.5589	0.6123
L	0.1112	0.1355

Table 6.4: Optimization results, problem 2, variant A.

Problem 2, Variant A

The second setup we are considering for optimization is problem 2, variant A. For this setup, the lift and drag components first have to be turned into the coordinate system which is aligned with the free stream direction and the signs have to be changed. Figure 6.22 shows the drag and lift defined in the correct directions. The drag is increasing both with the thickness of the profile and the angle of attack. The lift is increasing with increasing angle of attack and decreasing with increasing thickness. For low angles of attack, the lift is almost zero (resp. zero for zero angle of attack due to symmetry reasons) and is therefore almost constant for changing thickness.

The constrained optimization has been done again by using the objective function of problem 1, variant B with constraints given for the lift force:

$$J(\boldsymbol{\mu}) = 10\frac{D(\boldsymbol{\mu})}{D_{\text{ref}}} + \frac{\mu_1}{V_{\text{ref}}}. \tag{6.24}$$

Now we can interpret the resulting lift as desired quantity as we could easily reflect the setup on the x_1-axis to get a negative lift force. Therefore a desired interval for the resulting lift force is given by $L_{\max} = 0.13$, $L_{\min} = 0.11$. The unconstrained optimization is done using the objective function

$$J(\boldsymbol{\mu}) = 10\frac{D(\boldsymbol{\mu})}{L(\boldsymbol{\mu})} + \mu_1. \tag{6.25}$$

Here, the first term maximizes the lift-to-drag ratio (resp. minimizes the drag-to-lift ratio). The second term tries to keep the volume of the profile small. The results for the two strategies and cost functionals are shown in table 6.4. The unconstrained optimization results in a higher lift force but accepts a higher drag force. The constrained optimization finds a setup with a lift force close to the minimum lift allowed with the smallest possible value for the drag in the feasible region. Figure 6.23 shows the resulting pressure distributions for the optimal setups.

Problem 2, Variant B

The last setup we are considering for optimization is problem 2, variant B. As for the previous setup, the lift and drag components first have to be turned into the coordinate system which is aligned with the free stream direction and the signs have to be changed. We skip the presentation of the resulting lift and drag forces here.

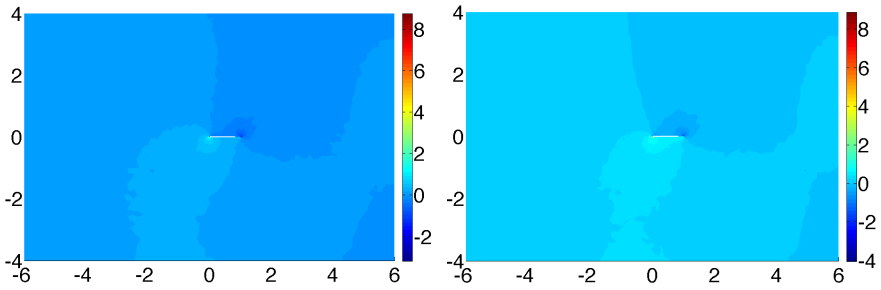


Figure 6.23: Optimal pressure distribution: constrained (left) and unconstrained (right) optimization procedure, problem 2, variant A.

	constrained	unconstrained
μ_1^{opt}	4	4
μ_2^{opt}	0.1342	0.15
μ_3^{opt}	1.8105	2.0
D	1.1932	0.6331
L	-0.0577	-0.0053

Table 6.5: Optimization results, problem 2, variant B.

The constrained optimization has been done again by using the objective function of problem 1, variant B with constraints for the lift:

$$J(\boldsymbol{\mu}) = 10\frac{D(\boldsymbol{\mu})}{D_{\text{ref}}} + \frac{\mu_1}{V_{\text{ref}}}. \tag{6.26}$$

We have to require a negative lift in this case. Therefore we define an interval in which the resulting lift should reside: $L_{\min} = -0.13$ and $L_{\max} = -0.05$. The unconstrained optimization is done now with the following objective function:

$$J(\boldsymbol{\mu}) = 10D(\boldsymbol{\mu}) + 50L(\boldsymbol{\mu}) + \mu_1. \tag{6.27}$$

The first term minimizes the drag while the second term maximizes the negative lift. It has to be kept in mind that the desired lift has a negative sign. The third term controls the volume of the profile. Table 6.5 shows the results of the optimizations. It becomes clear that the optimal settings are very different in this case for the two procedures. The weighting of the lift force in the unconstrained optimization is relatively small compared to the drag and therefore only a small negative lift is obtained. Consequently, the drag can be reduced much more compared to the constrained optimization where a certain value for the lift has to be obtained. Figure 6.24 shows the resulting pressure distributions for the optimal setups for both optimization variants.

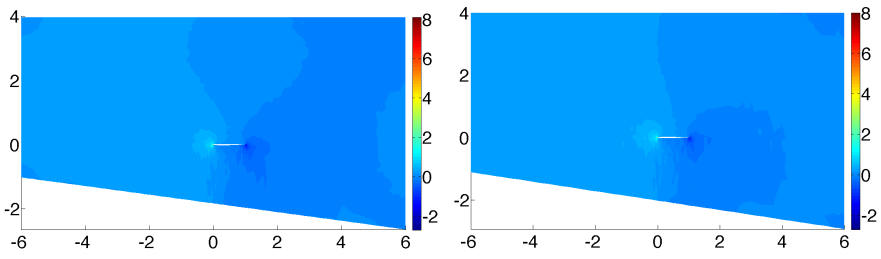


Figure 6.24: Optimal pressure distribution: constrained (left) and unconstrained (right) optimization procedure, problem 2, variant B.

7 The Reduced Basis Method for the Standard Stokes Equations: A Perspective

To conclude our presentation of the application of the reduced basis method to a number of flow models with increasing complexity, we will give a brief outlook on the solution of the standard Stokes equations (6.1) by the reduced basis method. Recalling the properties of the approximation of Stokes flow by a penalty method in the previous chapter, we can state several important simplifications of the resulting equation system compared to the standard Stokes system. The first benefit of the approximation of the Stokes flow by the penalty method is that it allows us to remain in the coercive case. The second benefit is the reduction of the system that makes the introduction of an additional solution space for the pressure unnecessary. The problems and difficulties that arise if we want to solve the full Stokes system instead of the penalty formulation shall be explained briefly in the following. Detailed explanations about the use of the RB method for Stokes flow can be found for example in [66], [67] and [69].

The Stokes equation system is non-coercive. The implications of this are that we can not longer rely on a coercivity constant for stability analysis and the computation of our a posteriori error bounds. Instead, we have to carry out a stability analysis based on the more general "Brezzi-Babuska" inf-sup constant

$$\beta(\boldsymbol{\mu}) = \inf_{q \in M_h} \sup_{\boldsymbol{w} \in Y_h} \frac{b(q, \boldsymbol{w}; \boldsymbol{\mu})}{\|\boldsymbol{w}\|_Y \|q\|_M} \tag{7.1}$$

to verify stability and well-posedness. Here, $b(q, \boldsymbol{w}; \boldsymbol{\mu})$ is the parameter dependent linear form (7.9) arising of the pressure term in the momentum equation in the weak form (see [15], [69]) and M_h and Y_h are the pressure respectively velocity test spaces. For the a posteriori error estimation, a second, different inf-sup constant, the "Babuska" constant, is necessary, see [24, 77]:

$$\overline{\beta(\boldsymbol{\mu})} = \inf_{\boldsymbol{w} \in Y_h} \sup_{\boldsymbol{v} \in Y_h} \frac{a(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{\mu})}{\|\boldsymbol{w}\|_Y \|\boldsymbol{v}\|_Y}, \tag{7.2}$$

where $a(\boldsymbol{w}, \boldsymbol{v}; \boldsymbol{\mu})$ is the parameter dependent bilinear form (7.8). The computation of the inf-sup constants is computationally more complex than the computation of the coercivity constant and therefore increases the offline computational times considerably.

The introduction of the additional variable, the pressure, and the proper treatment of this problem and the arising stability issues have to be considered both in the underlying FE approximation and in the RB approximation. For the FE approximation, the stability of the solution has to be ensured by the proper (stable) choice of approximation spaces for pressure and velocity (e.g. P1 - P2), or by applying alternative stabilization techniques (details can be found e.g. in [14], [15] and [62]). The arising equation system is thus enlarged considerably, first by considering an additional unknown, and second by the stable choice for the velocity approximation space, which has normally to be of higher order than the pressure space or an other approximation space used for scalar equations or e.g. the linear elasticity system.

This could be a benefit for the RB method, as the computational savings could be higher. But also for the RB method, an additional approximation space has to be introduced for the pressure and the corresponding velocity space has to be chosen in a way that assures stability. In this case, the choice of the RB pressure space is not complicated: It is built as usual of the pressure solutions (snapshots) for a number N of parameter samples. The choice for the velocity space is

more difficult. In addition to the velocity solutions for the N parameter samples, the space has to be enriched by other functions to ensure stability. This can be done for example by adding the N so-called supremizer solutions $\sigma_{\mathbf{k}}(\mu_{\mathbf{k}})$ which are obtained by solving the additional problem

$$\int_{\Omega}(\sigma_{\mathbf{k}} \cdot \mathbf{w} + \nabla\sigma_{\mathbf{k}} \cdot \nabla\mathbf{w}) = \int p_k \nabla \cdot \mathbf{w} \tag{7.3}$$

to the velocity space. Here, the solution $\sigma_{\mathbf{k}}$ and the test functions $\mathbf{w}$ both belong to the same Sobolev space ([65], [69]) This method, together with the fact that two approximation spaces are needed, leads to an increased size of the resulting equation system, as it is the case for the FE method (that means an increased online complexity) and also to an increased complexity in the offline computation of the RB basis functions.

In addition, it is useful to apply an orthonormalization procedure to the velocity and pressure basis functions of the RB system to control and reduce the condition number of the resulting system. The orthonormalization process is in this case not trivial, because special attention has to be paid not to spoil the stability qualities of the original set of basis functions. More details about suitable procedures can again be found in [69].

7.1 Problem Statement

The Stokes equations (6.1) in weak formulation read as follows:

$$\begin{cases} \mu \displaystyle\int_{\Omega_o(\boldsymbol{\mu})} \nabla\boldsymbol{u} \cdot \nabla\boldsymbol{v} - \int_{\Omega_o(\boldsymbol{\mu})} p\nabla \cdot \boldsymbol{v} = \int_{\Omega_o(\boldsymbol{\mu})} \boldsymbol{f} \cdot \boldsymbol{v} - \mu \int_{\partial\Omega_o^D(\boldsymbol{\mu})} \nabla\boldsymbol{g} \cdot \nabla\boldsymbol{v}, & \forall \boldsymbol{v} \in Y \\ \displaystyle\int_{\Omega_o(\boldsymbol{\mu})} q\nabla \cdot \boldsymbol{u} = -\int_{\partial\Omega_o^D(\boldsymbol{\mu})} q\nabla \cdot \boldsymbol{g} & \forall q \in M \end{cases} \tag{7.4}$$

For this formulations, the boundary conditions

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \Gamma_D, \tag{7.5}$$

$$\mu\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}} - p\boldsymbol{n} = \boldsymbol{0} \qquad \text{on } \Gamma_N. \tag{7.6}$$

for boundaries Γ_D with Dirichlet conditions and boundaries Γ_N with Neumann conditions were used. The compact formulation of the weak statement follows as

$$\begin{cases} a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) + b(p, \boldsymbol{v}; \boldsymbol{\mu}) = f(\boldsymbol{v}; \boldsymbol{\mu}) & \forall \boldsymbol{v} \in Y \\ -b(q, \boldsymbol{u}; \boldsymbol{\mu}) = g(q; \boldsymbol{\mu}) & \forall q \in M \end{cases} \tag{7.7}$$

with

$$a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu}) = \mu \int_{\Omega_o(\boldsymbol{\mu})} \nabla\boldsymbol{u} \cdot \nabla\boldsymbol{v} \tag{7.8}$$

$$b(p, \boldsymbol{v}; \boldsymbol{\mu}) = -\int_{\Omega_o(\boldsymbol{\mu})} p\nabla \cdot \boldsymbol{v} \tag{7.9}$$

$$f(\boldsymbol{v}; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} \boldsymbol{f} \cdot \boldsymbol{v} - \mu \int_{\partial\Omega_o^D(\boldsymbol{\mu})} \nabla\boldsymbol{g} \cdot \nabla\boldsymbol{v} \tag{7.10}$$

$$g(q; \boldsymbol{\mu}) = -\int_{\partial\Omega_o^D(\boldsymbol{\mu})} q\nabla \cdot \boldsymbol{g}. \tag{7.11}$$

The formulation on the reference domain Ω (2.1) finally follows with the methods of section 2.7.

7.2 Numerical Results

The rbMIT software package in the officially released version is not yet designed to handle the standard Stokes equations. As we explained in the previous sections, the solution of the standard Stokes system is computationally more costly and involves different procedures than the previous fluid models due to the non-coercivity. Nevertheless, the application of the RB method to Stokes flow around the geometric setups used in the previous chapters would be an interesting and necessary next step towards the solution of the full Navier-Stokes equations. We therefore calculated the RB approximation for the field solution for the setup of problem 2, variant A (see table 4.1) with a beta version of the rbMIT software under testing. This version is able to work with the standard Stokes system, but the necessary algorithms and procedures are not optimized yet and therefore the offline runtime (especially of the modified SCM algorithm) is still very high. A detailed investigation of the RB method for the Stokes system was thus beyond the scope of this work but is a promising perspective for further work with more complicated fluid models.

The underlying FE solution of this problem is based on a P2-P1 approximation [69]. Therefore the FE mesh is relatively coarse to keep the size of the underlying FE problem sufficiently small. The FE mesh with 1792 mesh elements and the affine decomposition can be found in appendix A. The affine decomposition consists of 98 subdomains and results in 65 affine terms. Note that the specification of the geometry and thus the affine decomposition is identical to the one used for the approximation of Stokes flow by a penalty method for the same problem. However, the resulting number of affine terms is more than doubled for the approximation of the standard Stokes system. The number of steps in the SCM algorithm has been restricted to $J_{\max} = 3$ to reduce the RB offline time. Nevertheless, the error estimate seems to be approximated sufficiently accurate, as the number of basis functions for the primal RB space is only $N_{pr} = 7$ which is less than for the corresponding problem in chapter 6. The reason for this behavior are the reduced variations in the field solutions due to the exact compliance with the continuity equation. The distribution of the greedy sample points for the primal RB space is depicted in figure 7.1. We note that as for most of the previous computations, the sample points form a contour region.

Figure 7.2 shows the velocity components, the pressure field and the vectors of the velocity field for $\boldsymbol{\mu} = (14, \pi/16)$. The boundary conditions are identical to the ones used in the previous chapter for the solution of the Stokes equations by a penalty method, see figure 6.1. We note that the continuity equation is now fulfilled perfectly and no artificial mass sinks and sources exist. In addition, the pressure field is smoother than the pressure fields obtained as a function of the velocity solution in the previous chapter, since we do no longer solve the governing PDE only for the velocity field, but also for the pressure field. The velocity fields are nearly antisymmetric and the influence of the angle of attack is clearly visible, while the influence of the angle of attack on the pressure field is slightly less articulated. Contrary to the potential flows investigated in chapter 4, only the front stagnation point moves on the lower part of the profile with increasing angle of

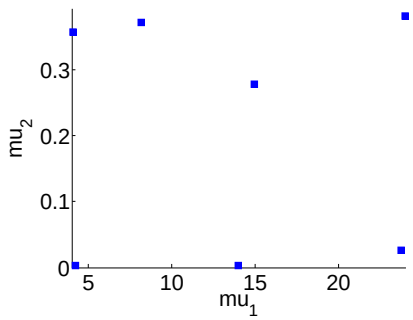


Figure 7.1: Sample points of the greedy primal RB space.

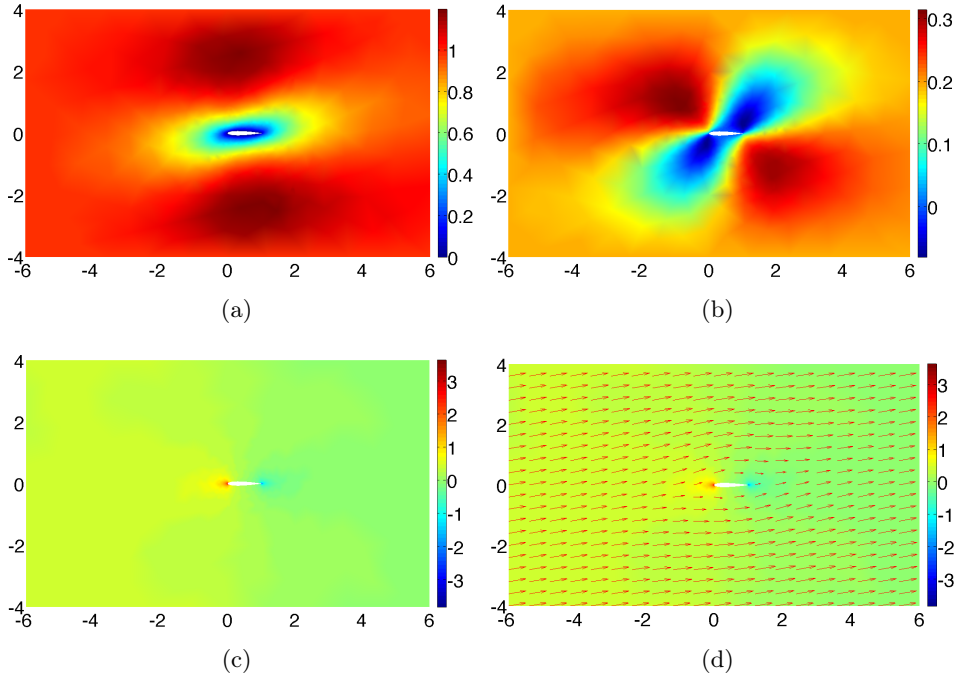


Figure 7.2: Velocity components in x_1-direction (a) and x_2-direction (b), pressure field (c) and velocity vectors (d) for $\boldsymbol{\mu} = (14, \pi/16)$.

attack, while the rear stagnation point remains fixed on the trailing edge since the flow field is fulfilling the Kutta condition.

Comparing the computational times in table 7.1 with the computational times for the corresponding problem in table 6.1, we note that the RB online times $t_{RB}^{\text{online},100}$ are comparable, although we note the doubling due to the increased number of affine terms. Due to memory reasons, the FE online time $t_{FE}^{\text{online},100}$ for this problem could not be determined. The FE and RB offline times are increased considerably since the complexity of the problem is higher than for the governing equations in chapter 6. Especially the RB offline time t_{RB}^{offline} is very high as on the one hand the beta version of the rbMIT software contains an SCM algorithm which is not yet optimized, and on the other hand the non-coercivity requires more expensive computational procedures, as explained above.

	P2VA
# elements mesh N_t	1792
# subdomains	98
affine terms Q_a	65
affine terms Q_f	0
# RB functions $N_{\max}$	7
steps SCM $J_{\max}$	3 (fixed)
t_{FE}^{offline}	4h 10min
t_{RB}^{offline}	28h 22min
$t_{RB}^{\text{online},100}$	0.0814s

Table 7.1: Summary or problem 2, variant A.

8 Summary and Outlook

In this thesis, we studied the reduced basis method for flows around racing car components. We first introduced the fundamentals of the reduced basis method for parametrized coercive elliptic PDEs. Special attention has been laid on the presentation of rigorous and sharp a posteriori error estimates and their efficient computation. The basic methodology for symmetric and compliant problems has been extended for noncompliant problems and nonsymmetric operators. In addition, the theory of affine mappings necessary for the transformation of parameter dependent geometries to parameter independent reference domains has been studied.

After an introduction to its use and capabilities, the rbMIT software package, a software designed especially for the approximation of parametrized PDEs with the RB method, was used here to apply the reduced basis method to different physical models for flows around parametrized airfoils, thanks to the parametrizations of the different geometries containing one to four parameters and describing variations in geometry, boundary conditions and flow characteristics. Physical models of increasing complexity allowed us to study the behavior and performance of the RB method for different underlying partial differential equations which subsequently required more extensions to the basic reduced basis methodology.

As first flow model we considered the non-viscous model of potential flows, governed by a symmetric scalar PDE. The examination of the automatic affine geometry decomposition and transformation to the reference geometry showed that this step, especially for complex curved geometries, is not trivial and should be guided in a careful way by the user. A considerably increased efficiency of the resulting RB approximation may result. Nevertheless, the automatization of this step by the software is a great simplification for the user. Comparing the distributions of the sample points for the different basis spaces, we noted a contour region distribution of the sample points for the majority of the setups, which could be confirmed also for the subsequent flow models. The a posteriori error estimates have proven to be extremely sharp for potential flows. The online speedup of the RB solution compared to the solution by the FE method was indeed very high for all setups.

For the second flow model we chose a class of thermal flows, described by a nonsymmetric scalar convection-diffusion equation for the distribution of e.g. the temperature. We considered not only the field solution but also the computation of scalar outputs and thus the application of a full primal-dual framework for noncompliant problems became necessary. The introduction of the Peclet number as a parameter resulted in a strong variation of the flow behavior for different values of this parameter. This resulted in higher numbers of basis functions in the primal and dual RB spaces, however, the size of the basis spaces remained relatively moderate. Admittedly, the size of the reduced basis spaces (and consequently offline and online computational time) has been increased even further if ground effects and thus small distances between different geometric objects were included in the geometric setup. The a posteriori error estimates were clearly less sharp for this flow model than for the potential flows but the online speedup (especially for the variants without ground effects) remained high.

The last flow model, governed by the Stokes equations, introduced viscosity to our problem. To reduce the size of the problem and to remain in the coercive case, we applied a penalty method to reduce the standard Stokes system to a system equivalent to the linear elasticity equations. In addition to the velocity field solution, we computed the lift and drag forces acting on the profile. The computation of the output functionals for the complete parameter domain has proven to be very fast and thus the RB method is well-suited for the shape optimization context. The a posteriori

error estimates, which are a crucial ingredient in optimization applications, were again sharper for this flow model than for the thermal flows. Finally, we optimized several geometric setups with respect to lift and drag. This has been done by first computing the lift and drag forces in a fine resolution for the complete parameter domain and then searching the optimal setting with a relatively simple search algorithm, which was absolutely feasible due to the approximation by the RB method. To conclude, we discussed the application of the RB method to the standard Stokes system in the non-coercive case which is more complex and computationally more expensive. We calculated the RB approximation for the field solution for an exemplary geometric setup with this flow model.

Comparing the performance of the reduced basis method for the different fluid models and geometric settings, we came to the conclusion that the RB method is very well suited to reduce the order of problems with varying thermo-physical and mechanical properties or flow properties. The RB method is able to provide an astonishingly high reduction of the problem, even if the variations of the solution behavior for different values of these parameters are very strong and the solutions for the extremal admissible parameter values possess a completely different character. If the problem consists of a parametrized geometry, the performance of the reduced basis method depends on the complexity of the geometry and the nature of the variations. If the variations can be translated to affine mappings (as it is the case for all examples in this thesis), the approximation of the solution with the RB method is simplified a lot. However, if the geometry is highly complex, that means that e.g. many different parts are interacting, many different curved surfaces or objects have to be considered or very small distances between the surfaces occur, the affine decomposition of the domain becomes very expensive and in some cases such a decomposition is not even possible. The methods proposed in this thesis are not applicable to e.g. the optimization of complex configurations with non-affine mappings (in this case some extensions and more complex techniques and approaches should be used).

For isolated parts of such a complex geometry which allow an affine representation, the performance of the RB method is very good, even with the high order curved surfaces of the airfoils considered in this work. The time necessary for an optimization of such simplified setups is considerably decreased by the use of the RB method. If the user makes some effort in the advantageous specification of the parametrized geometry, large economizations in the offline times can be obtained. The automatic domain decomposition process implemented in the rbMIT software is then a great support and simplification for the user, but the flexibility of the software would also allow the usage of other external domain decomposition algorithms, which may be better suited for a specific problem. The use of the rbMIT software package in the high-level mode is very straightforward and needs only a short training period. For the application to more complex systems however, a certain degree of experience is beneficial to better understand and predict the behavior both of the affine decomposition algorithm and the reduced basis generation process together with the possible difficulties and obstacles. The generality of the low-level kernel makes this part of the software applicable to a wide variety of setups and PDEs.

The reduced basis method and its application to the shape optimization of racing car components and other problems and questions in fluid dynamics is a very interesting and broad field. To continue the work that we started in this thesis, the detailed investigation of the RB method for the standard Stokes system would be a next step. The last step in our sequence of flow models would be the application of the RB method to the Navier-Stokes equations for the parametrized airfoils. This would need the extension of the RB methodology for the standard Stokes equations to nonlinear systems, see e.g. [19, 76, 60] and should be considered as a very challenging task. In addition, the investigation of other, more realistic parametrizations of the airfoils (e.g. nonsymmetric profiles or other parametrizations than the NACA-4-digit family) and the inclusion of the simplified car geometry to study of the interactions between, for example, the rear wings, the car and the ground would be interesting and natural continuations of this work.

A Domain Decompositions and Finite Element Meshes

A.1 Potential Flows

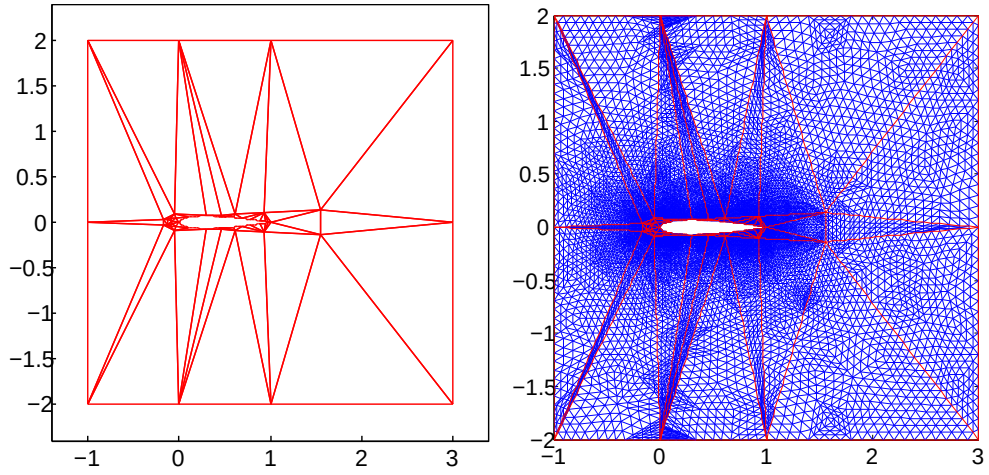


Figure A.1: Final geometry decomposition and FE mesh, problem 1, variant A, small domain.

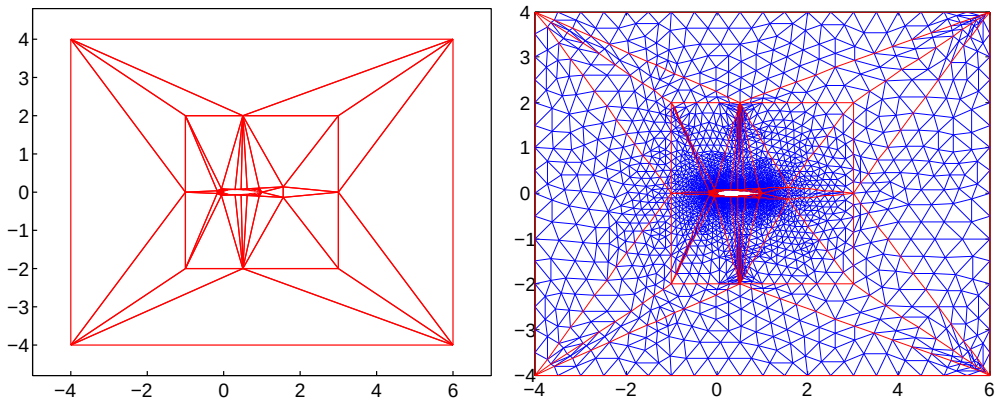


Figure A.2: Final geometry decomposition and FE mesh, problem 1, variant A, embedded domain.

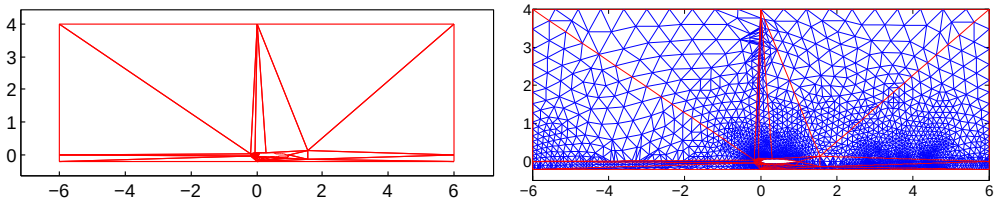


Figure A.3: Final geometry decomposition and FE mesh, problem 1, variant B, big domain.

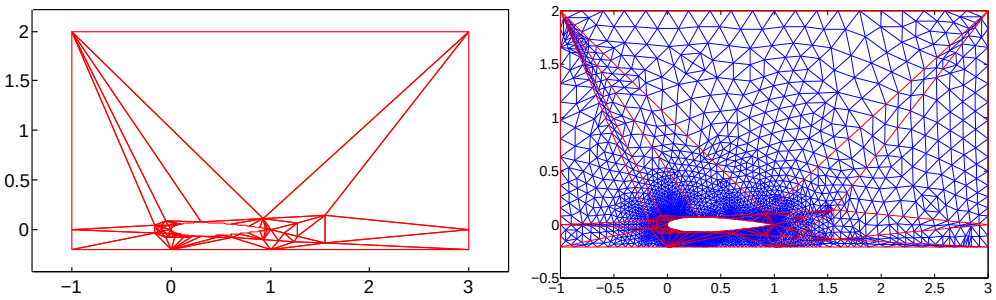


Figure A.4: Final geometry decomposition and FE mesh, problem 1, variant B, small domain.

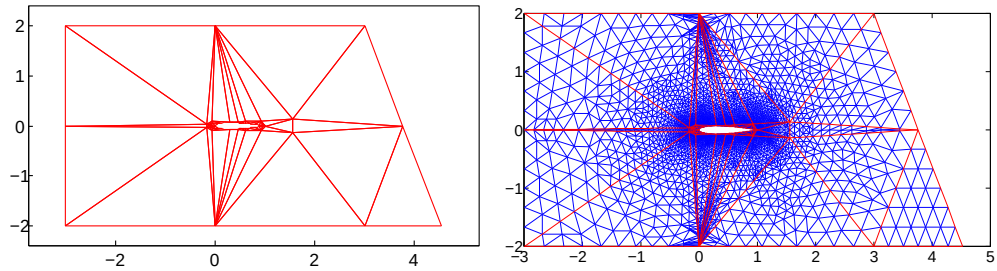


Figure A.5: Final geometry decomposition and FE mesh, problem 2, variant A, realization (a).

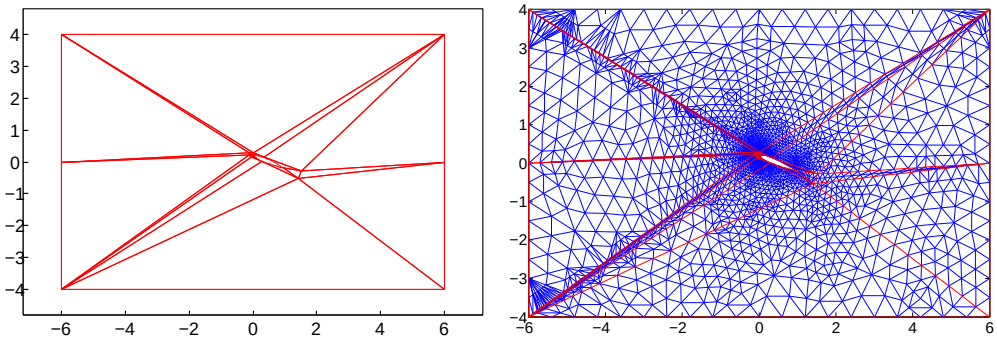


Figure A.6: Final geometry decomposition and FE mesh, problem 2, variant A, realization (b).

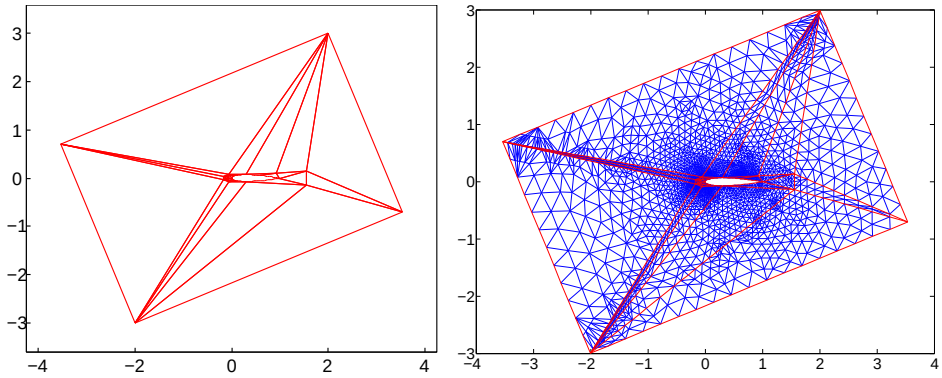


Figure A.7: Final geometry decomposition and FE mesh, problem 2, variant A, realization (c).

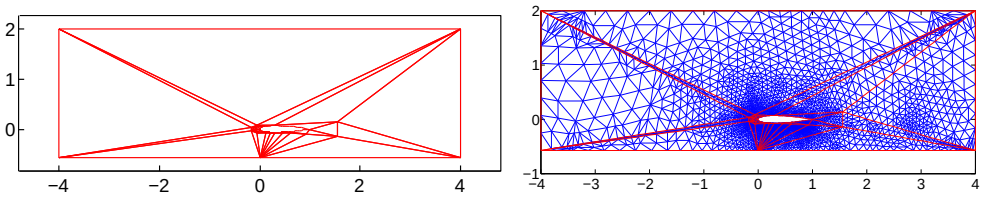


Figure A.8: Final geometry decomposition and FE mesh, problem 2, variant B, realization (a).

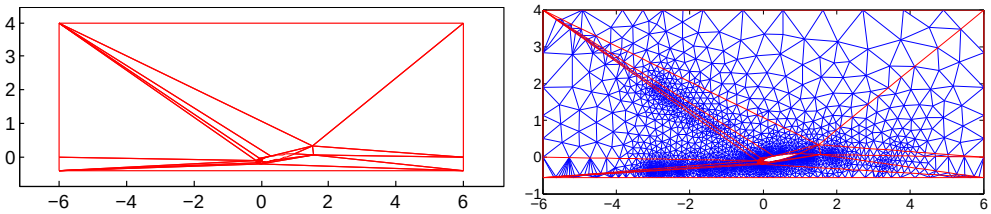


Figure A.9: Final geometry decomposition and FE mesh, problem 2, variant B, realization (b).

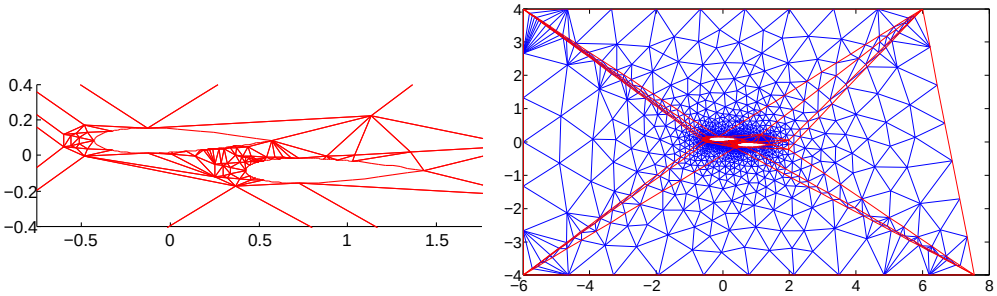


Figure A.10: Final geometry decomposition (detail) and FE mesh, problem 3.

A.2 Thermal Flows

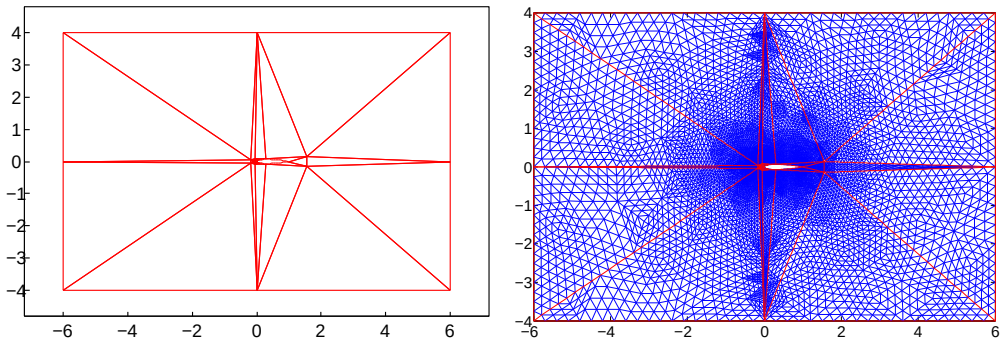


Figure A.11: Final geometry decomposition and FE mesh, problem 1, variant A.

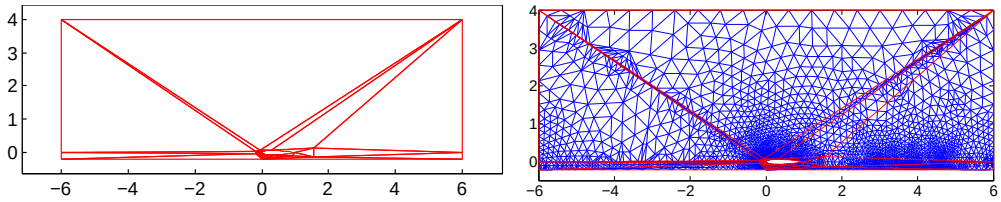


Figure A.12: Final geometry decomposition and FE mesh, problem 1, variant B.

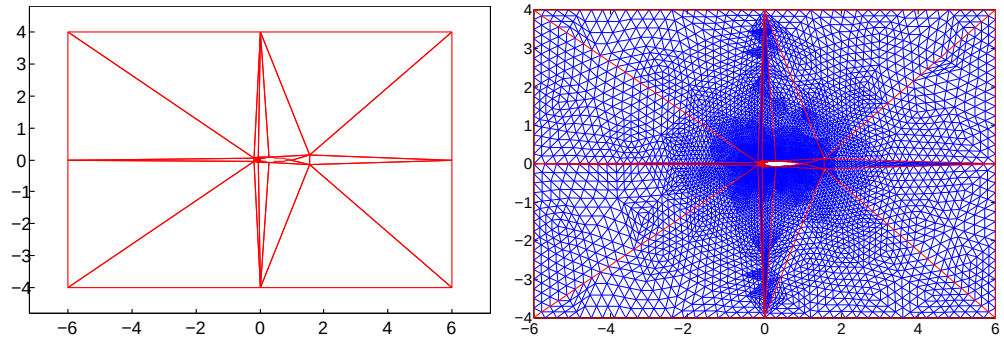


Figure A.13: Final geometry decomposition and FE mesh, problem 2, variant A.

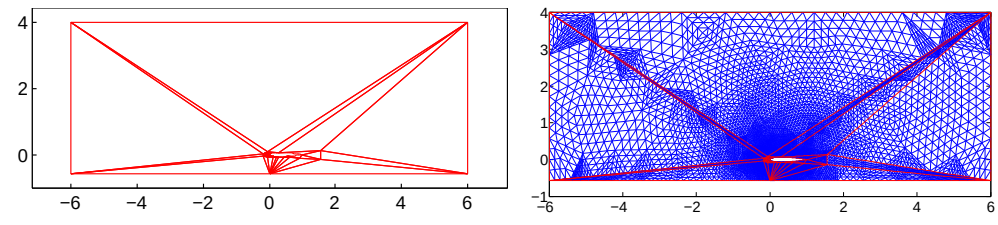


Figure A.14: Final geometry decomposition and FE mesh, problem 2, variant B.

A.3 Stokes Flow by a Penalty Method

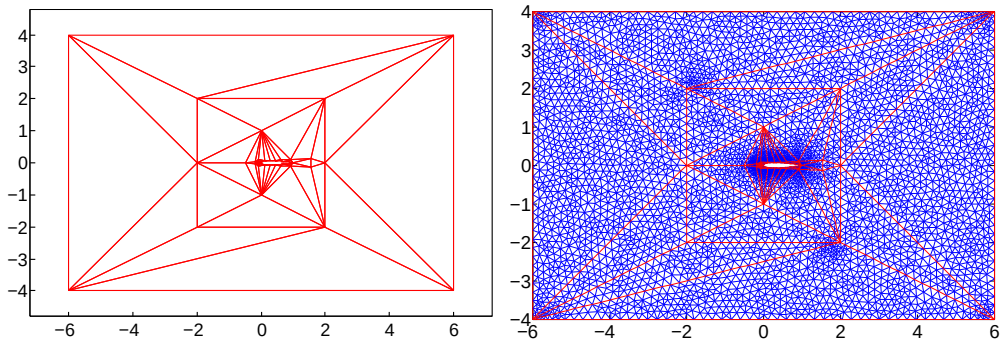


Figure A.15: Final geometry decomposition and FE mesh, problem 1, variant A.

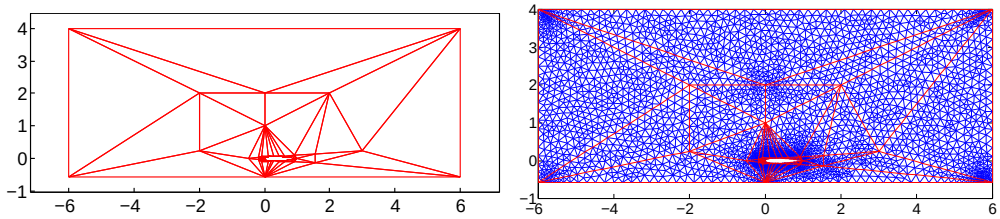


Figure A.16: Final geometry decomposition and FE mesh, problem 1, variant B.

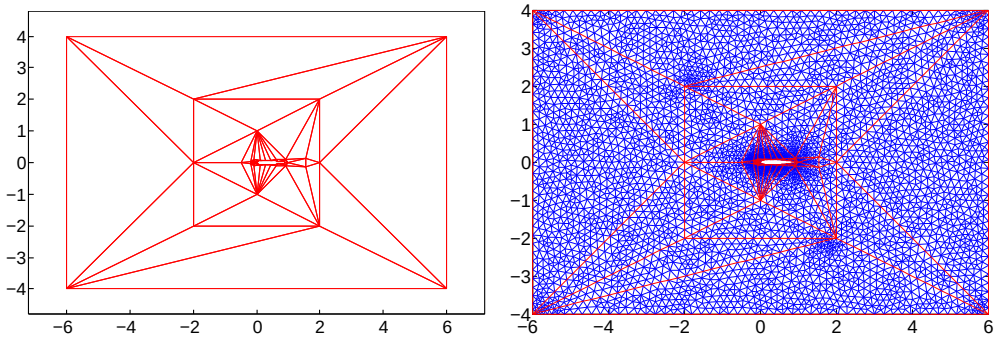


Figure A.17: Final geometry decomposition and FE mesh, problem 2, variant A.

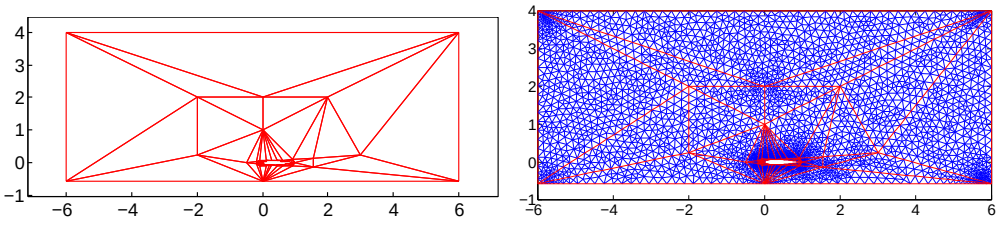


Figure A.18: Final geometry decomposition and FE mesh, problem 2, variant B.

A.4 Stokes Flow (Standard Stokes System)

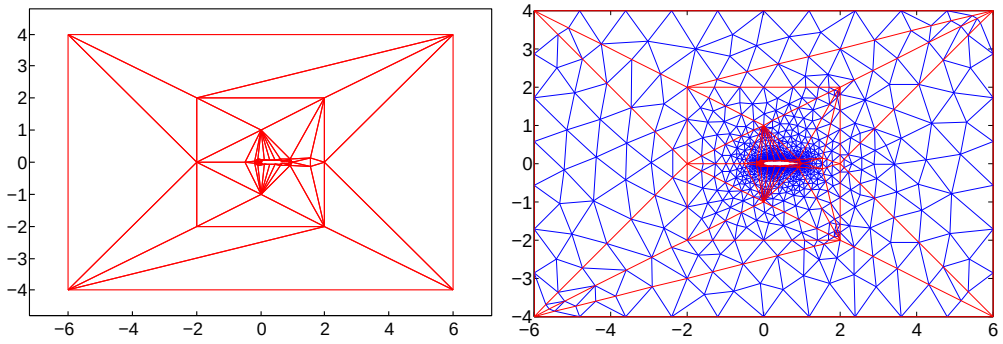


Figure A.19: Final geometry decomposition and FE mesh, problem 2, variant A.

Bibliography

[1] I.H. Abbott and A.E. von Doenhoff. *Theory of Wing Sections*. McGraw-Hill Book Company, Inc., 1949.

[2] B.O. Almroth, P. Stern, and F.A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16:525-528, 1978.

[3] V.S. Arpaci. *Conduction heat transfer*. Addison-Wesley, 1966.

[4] V.S. Arpaci and P.S. Larsen. *Convection heat transfer*. Prentice Hall, 1984.

[5] Reduced Basis at MIT. http://augustine.mit.edu/methodology.htm, 2008.

[6] E. Balmes. Parametric families of reduced finite element models. Theory and applications. *Mechanical Systems and Signal Processing*, 44(170):283-301, 1996.

[7] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus de l'Académie des Sciences Paris*, Sér I 339:667-672, 2004.

[8] A. Barrett and G. Reddien. On the reduced basis method. *Zeitschrift für Angewandte Mathematik und Mechanik*, 75(7):543-549, 1995.

[9] J.J. Bertin. *Aerodynamics for Engineers*. Prentice Hall, 2002.

[10] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag New York, 1991.

[11] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large scale systems with high-dimensional parametric input space. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, AIAA Paper 2007-2049, 2007.

[12] G.F. Carey and R. Krishnan. Penalty approximation of Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 35:169-206, 1982.

[13] L. Dedè. *Adaptive and Reduced Basis Methods for Optimal Control Problems in Environmental Applications*. PhD thesis, Politecnico di Milano, 2008.

[14] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, Inc, 2003.

[15] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford university Press, 2005.

[16] J.P. Fink and W.C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *Zeitschrift für Angewandte Mathematik und Mechanik*, 63:21-28, 1983.

[17] J.P. Fink and W.C. Rheinboldt. Local error estimates for parametrized non-linear equations. *SIAM Journal on Numerical Analysis*, 22:729-735, 1985.

[18] P.L. Gould. *Introduction to Linear Elasticity*. Springer-Verlag New York, second edition, 1994.

[19] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis*, 41(3):575-605, 2007.

[20] W. Hucho. *Aerodynamik des Automobils*. Vieweg Verlag, 2005.

[21] T.J.R. Hughes, W.K. Liu, and A. Brooks. Finite element analysis of incompressible viscous flows by the penalty function formulation. *Journal of Computational Physics*, 30:1-60, 1979.

[22] D.B.P. Huynh, N.C. Nguyen, G. Rozza, and A.T. Patera. *Documentation for rbMIT Software: I. Reduced Basis (RB) for Dummies*. ©Massachusetts Institute of Technology, 2007. available at http://augustine.mit.edu together with the rbMIT Software package.

[23] D.B.P. Huynh, N.C. Nguyen, G. Rozza, and A.T. Patera. rbMIT Software. http://augustine.mit.edu/methodology/methodology_rbmit_system.htm. ©*Massachusetts Institute of Technology, Cambridge*, 2007.

[24] D.B.P. Huynh, G. Rozza, S. Sen, and A.T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus de l'Académie des Sciences Paris*, Sér I 345:473-478, 2007.

[25] K. Ito and S.S. Ravindran. A reduced basis method for control problems governed by PDEs. *International Series of Numerical Mathematics*, 126:153-168, 1998.

[26] K. Ito and S.S. Ravindran. A reduced-order method for simulation and control of fluid flow. *Journal of Computational Physics*, 143(2):403-425, 1998.

[27] K. Ito and S.S. Ravindran. Reduced basis method for optimal control of unsteady viscous flows. *International Journal of Computational Fluid Dynamics*, 15:97-113, 2001.

[28] K. Ito and J.D. Schroeter. Reduced order feedback synthesis for viscous incompressible flows. *Mathematical And Computer Modelling*, 33(1-3):173-192, 2001.

[29] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA Paper*, 95-1729, 1995.

[30] W.M. Lai, D. Rubin, and E. Krempl. *Introduction to Continuum Mechanics*. Butterworth-Heinemann, third edition, 1993.

[31] M.Y. Lin Lee. Estimation of the error in the reduced-basis method solution of differential algebraic equations. *SIAM Journal on Numerical Analysis*, 28:512-528, 1991.

[32] S.-Y. Lin, Y.-S. Chin, and T.-M. Wu. A modified penalty method for Stokes equations and its application to Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 16(1):1-19, 1995.

[33] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag Berlin, 1971.

[34] L. Machiels, Y. Maday, I.B. Oliveira, A.T. Patera, and D.V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *Comptes Rendus de l'Académie des Sciences Paris*, Sér I 331(2):153-158, 2000.

[35] Y. Maday, L. Machiels, A.T. Patera, and D.V. Rovas. Blackbox reduced-basis output bound methods for shape optimization. In *Proceedings of the 12th international Domain Decomposition Conference*, 2000.

[36] Y. Maday, A.T. Patera, and D.V. Rovas. A blackbox reduced-basis output bound method for noncoercive linear problems. In J.L Lions D. Cioranescu, editor, *Nonlinear partial differential equations and their applications*, Collège de France Seminar, vol. XIV, pages 535–569. Elseviever, Amsterdam, 2002.

[37] Y. Maday, A.T. Patera, and G. Turnici. Global a priori convergence theory for reduced basis approximation of single-parameter symmetric coercive elliptic partial differential equations. *Comptes Rendus de l'Académie des Sciences Paris*, Sér I 335(3):289-294, 2002.

[38] Y. Maday, A.T. Patera, and G. Turnici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing*, 17(1-4):437-446, 2002.

[39] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids.* Oxford University Press, 2001.

[40] N.C. Nguyen. *Reduced-Basis approximation and A Posteriori Error Bounds for Nonaffine and Nonlinear Partial Differential Equations: Applicaton to Inverse Analysis.* PhD thesis, National University of Singapore, 2005.

[41] N.C. Nguyen, K. Veroy, and A.T. Patera. Certified real-time solution of parametrized partial differential equations. In S. Yip, editor, *Handbook of Materials Modeling*, pages 1523–1558. Springer, 2005.

[42] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer-Verlag New York, 1999.

[43] A.K. Noor. Recent advances in reduction methods for nonlinear problems. *Computers & Structures*, 13:31-44, 1981.

[44] A.K. Noor. On making large nonlinear problems small. *Computer Methods in Applied Mechanics and Engineering*, 34:955-985, 1982.

[45] A.K. Noor, C.M. Andresen, and J.A. Tanner. Exploiting symmetries in the modeling and analysis of tires. *Computer Methods in Applied Mechanics and Engineering*, 63:37-81, 1987.

[46] A.K. Noor, C.D. Balch, and M.A. Shibut. Reduction methods for non-linear steady-state thermal analysis. *International Journal for Numerical Methods in Engineering*, 20:1323-1348, 1984.

[47] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18(4):455-462, 1980.

[48] A.K. Noor and J.M. Peters. Multiple-parameter reduced basis technique for bifurcation and post-buckling analysis of composite plates. *International Journal for Numerical Methods in Engineering*, 19:1783-1803, 1983.

[49] A.K. Noor and J.M. Peters. Recent advances in reduction methods for instability analysis of structures. *Computers & Structures*, 16:67-80, 1983.

[50] A.K. Noor, J.M. Peters, and C.M. Andersen. Mixed models and reduction techniques for large-rotation nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 44:67-89, 1984.

[51] J.T. Oden, N. Kikuchi, and Y.J. Song. Penalty-finite element methods for the analysis of Stokesian flows. *Computer Methods in Applied Mechanics and Engineering*, 31:297-329, 1982.

[52] I.B. Oliveira and A.T. Patera. Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations. *Optimization and Engineering*, 8:43-65, 2007.

[53] R. L. Panton. *Incompressible Flow*. John Wiley & Sons, Inc., 3rd edition edition, 2005.

[54] A.T. Patera and G. Rozza. Reduced basis approximation and error bounds for potential flows in parametrized geometries. *In preparation*, 2008.

[55] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. to appear in MIT Pappalardo Graduate Monographs in Mechanical Engineering, Copyright MIT 2006. Version 1.0.

[56] J.S. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):777-786, 1989.

[57] T.A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487-496, 1985.

[58] T.A. Porsching and M.Y. Lin Lee. The reduced-basis method for initial value problems. *SIAM Journal on Numerical Analysis*, 24:1277-1287, 1987.

[59] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations. *Journal of Fluids Engineering*, 124:70-80, 2002.

[60] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. *Numerical Methods for PDEs*, 23(4):923-948, 2007.

[61] A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis methods for optimal control of advection-diffusion problems. In W. Fitzgibbon, R. Hoppe, and J. Periaux et al., editors, *Advances in Numerical Mathematics*, pages 193–216. Moscow, Russia and Houston, USA, 2007.

[62] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag Berlin, 1997.

[63] W.C. Rheinboldt. On the theory and error estimation of the reduced basis method for multi-parameter problems. *Nonlinear Analysis, Theory, Methods and Applications*, 21(11):849-858, 1993.

[64] D.V. Rovas. *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, 2003.

[65] G. Rozza. Optimization, control and shape design of an arterial bypass. *International Journal Numerical Methods in Fluids*, 47(10-11):1411-1419, 2005.

[66] G. Rozza. Reduced basis methods for Stokes equations in domains with non-affine parametric dependence. *Computing and Visualization in Science*, doi: 10.1007/s00791-006-0044-7, 2006.

[67] G. Rozza. *Shape Design by Optimal Flow Control and Reduced Basis Techniques: Applications to Bypass Configurations in Haemodynamics*. PhD thesis, Ecole Polytechnique Féderale de Lausanne, November 2005.

[68] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3), 2008.

[69] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer methods in applied mechanics and engineering*, 196:1244-1260, 2007.

[70] W.E. Schiesser and C.A. Silebi. *Computational Transport Phenomena: Numerical Methods for the Solution of Transport Problems*. Cambridge University Press, 1997.

[71] W. Schröder. *Fluidmechanik - Aachener Beiträge zur Strömungsmechanik, Band 7*. Wissenschaftsverlag Mainz in Aachen, 2004.

[72] S. Sen, K. Veroy, D.B.P. Huynh, S. Deparis, N.C. Nguyen, and A.T. Patera. "Natural norm" a posteriori error estimators for reduced basis approximations. *Journal of Computational Physics*, 217:37-62, 2006.

[73] R. Temam. *Navier-Stokes Equations. Theory and Numerical Analysis*. North-Holland, Amsterdam, third edition, 1984.

[74] T. Tonn and K. Urban. A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies. In *ECCOMAS CFD 2006 proceedings*. P. Wesseling, E. Oñate and J. Périaux (Eds), TU Delft, The Netherlands, 2006.

[75] K. Veroy. *Reduced Basis Methods Applied to Problems in Elasticity: Analysis and Applications*. PhD thesis, Massachusetts Institute of Technology, 2003.

[76] K. Veroy and A.T. Patera. Certified real-time solutions of the parametrized steady incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 47:773-788, 2005.

[77] K. Veroy, C. Prud'homme, D.V. Rovas, and A.T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *Proceedings of the 16th AIAA computational fluid dynamics conference*, Paper 2003-3847, 2003.